

Fachhochschule Köln
University of Applied Sciences Cologne
Abteilung Gummersbach

Fachbereich Informatik

Diplomarbeit
zur Erlangung
des Diplomgrades
Diplom-Wirtschaftsinformatiker (FH)
in der Fachrichtung Informatik

„Entwicklung einer datenbankgestützten Web-Anwendung
zur Erfassung, Verfolgung und Dokumentation von Kundenwünschen und Fehlern“

Erstprüfer:	Prof. Dr. Heide Faeskorn-Woyke
Zweitprüfer:	Dipl. Ing. Andreas Braun
vorgelegt am:	16.01.2002
von:	Andreas Hurtmann
aus:	Am Sandberg 28 51643 Gummersbach
E-Mail:	Andreas.Hurtmann@web.de
Matr.-Nr.:	11014698 13

Inhaltsverzeichnis

ABBILDUNGSVERZEICHNIS:	4
TABELLENVERZEICHNIS:	5
1 VORWORT	6
1.1 ENTWICKLUNG EINER DATENBANKGESTÜTZTEN WEB-ANWENDUNG ZUR ERFASSUNG, VERFOLGUNG UND DOKUMENTATION VON KUNDENWÜNSCHEN UND FEHLERN	6
1.2 AUFGABENSTELLUNG	8
1.3 AFT ATLAS FAHRZEUGTECHNIK GMBH	9
2 GRUNDLAGEN	12
2.1 INTRANET	12
2.2 HYPER TEXT MARKUP LANGUAGE (HTML)	14
2.2.1 Definition: HTML	14
2.2.2 HTML-Versionshistorie	16
2.2.3 Grundstruktur einer HTML-Datei	16
2.2.4 Definition von Regeln in einem HTML-Dokument	17
2.3 JAVASCRIPT	18
2.3.1 Was ist JavaScript?	18
2.3.2 Übersicht der Browserunterstützung	18
2.3.3 JavaScript ist nicht Java	19
2.3.4 Wo wird JavaScript definiert?	20
2.4 PHP	21
2.4.1 Wie arbeitet PHP?	22
2.4.2 Historie von PHP	22
2.4.3 Wie sieht ein PHP-Skript aus?	23
2.5 MYSQL	26
2.5.1 Definition: MySQL	26
2.5.2 Leistungsmerkmale von MySQL	28
2.5.3 Einschränkungen bei MySQL	29
3 INSTALLATION UND KONFIGURATION EINES LAMP-SYSTEM	32
3.1 SUSE LINUX 6.3	32
3.2 APACHE 1.3.20	36
3.3 MYSQL 3.23.40	38
3.4 PHP 4.06	42
4 SOFTWARE ENTWICKLUNG	45
4.1 DIE PLANUNGSPHASE – PHASE 1	45
4.2 DIE DEFINITIONSPHASE – PHASE 2	47
4.2.1 Funktionsbaum	52
4.2.2 Datenflussdiagramm	55
4.2.3 ER-Diagramm	58
4.3 DIE ENTWURFSPHASE – PHASE 3	61
4.4 DIE IMPLEMENTIERUNGSPHASE – PHASE 4	62
4.4.1 Prinzipien der Implementierung	63
4.4.2 Ziele der Implementierung	63
4.5 DIE ABNAHME- UND EINFÜHRUNGSPHASE – PHASE 5	64
4.5.1 Die Abnahmephase	64
4.5.2 Die Einführungsphase	65
4.6 DIE WARTUNGS- & PFLEGEPHASE – PHASE 6	65
5 SOFTWARE-QUALITÄTSSICHERUNG	68
5.1 DEFINITION: QUALITÄT UND SOFTWARE-QUALITÄT	68
5.2 QUALITÄTSMODELLE	69
5.3 QUALITÄTSZIELBESTIMMUNG	70

5.3.1	Qualitätsstufe	71
5.3.2	Qualitätsanforderungen	71
5.3.3	Qualitätszielbestimmung des AFT-DataLogger	71
6	ANWENDUNG: AFT- DATALOGGER.....	75
6.1	DIE BENUTZERRECHTE IM SYSTEM	76
6.2	DIE BENUTZERRECHTVERGABE	76
6.3	ALLGEMEINER SEITENAUFBAU	77
6.4	STARTPUNKT „YAWOHL!“	78
6.5	LOGIN	78
6.6	HAUPTMENU	79
6.6.1	Kundenwunsch/AFT-Produktfehler erfassen	81
6.6.2	Administration	85
6.6.3	Produktmanager	92
6.6.4	Bearbeiter	97
6.6.5	Passwort ändern	102
6.7	DATENBANKZUGRIFF	102
7	EINGESETZTE SOFTWARE	110
7.1	PHPMYADMIN	110
7.2	PHPED	112
7.3	ERWIN 3.52	112
8	KRITISCHE WÜRDIGUNG UND AUSBLICK	115
	ANHANG A – LITERATURVERZEICHNIS	116
	ANHANG B – QUELLCODE-AUSZUG	118
	SQL-STATEMENTS ZUM ANLEGEN DER DATENBANKSTRUKTUR IN MYSQL:	118
	SOURCECODE DES AFT-DATALOGGER: LOGIN_CHECK.PHP	124
	SOURCECODE DES AFT-DATALOGGER: ERFASSER_SPEICHERN.PHP	128
	ANHANG C – INHALT DER CD-ROM	135

Abbildungsverzeichnis:

Abbildung 1: Elemente eines Intranets	14
Abbildung 2: So entsteht eine dynamische Website PHP	22
Abbildung 3 : HTML-Datei mit einer PHP-Funktion vor dem Parsen	24
Abbildung 4: HTML-Datei mit einer PHP Funktion nach dem Parsen	24
Abbildung 5: Auswahl der Netzwerkkarte	34
Abbildung 6: Netzwerkgrundkonfiguration	35
Abbildung 7: Rechnernamen und Domäne	35
Abbildung 8 : Der Apache wurde erfolgreich installiert!	37
Abbildung 9: Der MySQL-Server läuft	41
Abbildung 10: Beispiel eines Funktionsbaumes	52
Abbildung 11: Funktionsbaum des AFT-DataLoggers	54
Abbildung 12: Datenfluß-Diagramm des AFT-DataLoggers	57
Abbildung 13: Das ER-Diagramm des AFT-DataLogger	60
Abbildung 14: Einflussfaktoren auf die Software-Architektur	62
Abbildung 15: Allgemeiner Seitenaufbau des AFT-DataLogger	77
Abbildung 16: Login des AFT-DataLogger (1)	79
Abbildung 17: Login des AFT-DataLogger (2)	79
Abbildung 18: Das Hauptmenu des AFT-DataLogger für einen User der alle Berechtigungen besitzt	80
Abbildung 19: Hauptmenu des AFT-DataLogger für einen User mit 'Bearbeiter'-Rechten	80
Abbildung 20: Erfassung der Allgemeinen Daten	81
Abbildung 22: Erfassungsdaten der Produktdaten	83
Abbildung 23: Erfassungsübersicht	85
Abbildung 24: Hinweis auf Speicherung der Daten	85
Abbildung 25: Hauptmenu des Administrators	86
Abbildung 26: Benutzerübersicht	86
Abbildung 27: Benutzerdaten ändern	87
Abbildung 28: Benutzer löschen	87
Abbildung 29: Neuen Benutzer anlegen	88
Abbildung 30: Produktübersicht	89
Abbildung 31: Anlegen eines neuen Produktes bei bekanntem Produktnamen	90
Abbildung 32: Anlegen eines neuen Produktes bei unbekanntem Produktnamen	90
Abbildung 33: Projektnummer zuordnen	92
Abbildung 34: Menu des Produktmanagers	93
Abbildung 35: Übersicht über alle Datensätze	93
Abbildung 36: Übersicht der in Bearbeitung befindlichen Datensätze	94
Abbildung 37: Übersicht der offenen Datensätze	95
Abbildung 38: Datensatz zuordnen	96
Abbildung 39: Menü des Bearbeiters	97
Abbildung 40: Gesamtansicht der Bearbeitungsseite	99
Abbildung 41: Fehlerlösung/Erledigung Kundenwunsch	100
Abbildung 42: Passwortänderung	102
Abbildung 43: Hauptmenü von PHPMyAdmin	111
Abbildung 44: Tabellenübersicht der Datenbank „DataLogger“	111

Abbildung 45: Möglichkeiten des Datenexports einer Tabelle	111
Abbildung 46: PHPed	112
Abbildung 47: Startbildschirm von ERwin.....	113
Abbildung 48: Programmoberfläche von ERwin.....	113
Abbildung 49: Preferences zur Einstellung der Krähenfussnotation	114

Tabellenverzeichnis:

Tabelle 1: HTML-Versionshistorie	16
Tabelle 2: JavaScript Unterstützung der Webbrowser	18
Tabelle 3: Unterschiede Java – Javascript.....	19
Tabelle 4: PHP-Versionen.....	23
Tabelle 5: Datentypen in PHP	25
Tabelle 6 : URL's zum Downloaden von MySQL	39
Tabelle 7 : Beispiel für eine Gliederung eines Lastenheftes	46
Tabelle 8: Vor- und Nachteile eines Funktionsbaumes.....	53
Tabelle 9: Beziehungen zwischen Entitäten	59
Tabelle 10: Grafische Darstellung in ERwin.....	59
Tabelle 11: Beziehungen zwischen Entitäten in ERwin	59
Tabelle 12: Software-Qualitätsmerkmale nach DIN ISO 9126.....	70
Tabelle 13: Qualitätszielbestimmung des AFT-DataLogger	74
Tabelle 14: Erfassungsfelder der Allgemeinen Daten	81
Tabelle 15: Erfassungsfelder der Kundendaten	82
Tabelle 16: Erfassungsfelder der Kundendaten	83
Tabelle 17: Erfassungsdaten der Produktdaten	84
Tabelle 18: Datenbankzugriffe nach Dateien.....	109

1 Vorwort

1.1 *Entwicklung einer datenbankgestützten Web-Anwendung zur Erfassung, Verfolgung und Dokumentation von Kundenwünschen und Fehlern*

Die vorliegende Diplomarbeit entstand an der Fachhochschule Köln, Abteilung Gummersbach, im Fachbereich Informatik in Zusammenarbeit mit der Firma AFT Atlas Fahrzeugtechnik GmbH in Werdohl. Sie besteht aus dieser schriftlichen Ausarbeitung und einer CD-ROM, die neben dem Sourcecode des entwickelten Produktes das Pflichtenheft enthält, das in enger Zusammenarbeit mit meinen Betreuern erstellt wurde.

Ziel dieser Diplomarbeit ist die Entwicklung eines Software-Produktes, welches in einem firmeninternen Intranet-System zum Einsatz kommt. Es werden die eingesetzten Techniken sowie die Phasen einer Software-Entwicklung und die Software-Qualitätssicherung erläutert. Zum Abschluss folgt die Vorstellung des fertigen Software-Produktes.

Neben der reinen Software-Entwicklung besteht die Aufgabe darin, einen eigens dafür zur Verfügung stehenden Webserver einzurichten. Die Firma AFT (AFT Atlas Fahrzeugtechnik GmbH) ist Microsoft Certified Solution Partner und setzt für das bestehende Intranet einen IIS-Webserver¹ und einen MS-SQL Server² von Microsoft ein. Für die Diplomarbeit soll ein anderer Weg gewählt werden, um das Software-Produkt in das Intranet einzubinden. Die Wahl fällt auf das im Internet sehr weit verbreitete LAMP-System. LAMP ist eine Abkürzung und steht für **L**inux³, **A**pache⁴, **M**ySQL⁵ und **P**HP⁶. Alle vier Komponenten sind lizenzfreie Freeware-Produkte. Ein LAMP-System ist in Punkto Stabilität und Geschwindig-

¹ IIS = Internet Information Server, Webserver von Microsoft

² MS-SQL Server = Microsoft SQL Server. Komplexes und professionelles Datenbanksystem von Microsoft

³ Linux = Open Source Betriebssystem auf der Basis von Unix. Linux hat im Bereich der Webserver eine große Verbreitung gefunden

⁴ Apache = Ein lizenzfreier Open Source Web-Server, der auf viele Plattformen portiert wurde

⁵ MySQL = Freies und kostenloses Datenbanksystem, das einen großen Funktionsumfang auf einfachem Niveau realisiert

⁶ PHP = **P**HP **H**ypertext **P**reprocessor. PHP zeichnet sich durch großen Funktionsumfang und Plattformunabhängigkeit aus

keit mit einem Microsoft-System vergleichbar. Die Installation und Konfiguration eines LAMP-Systems ist ebenfalls Bestandteil dieser Diplomarbeit.

Es folgt eine kurze Kapitelübersicht:

In Kapitel 2 werden die Grundlagen der eingesetzten Techniken (HTML⁷, JavaScript und PHP) sowie der Begriff Intranet und die Datenbank MySQL erläutert.

In Kapitel 3 wird die Installation und Konfiguration eines LAMP-System beschrieben. Ein LAMP-System stellt einen stabilen und schnellen Webserver mit integrierter PHP-Funktionalität und einem Datenbanksystem zur Verfügung.

Kapitel 4 beschäftigt sich mit der Software-Entwicklung. Ein Software-Produkt muss genau nach den gewünschten Ansprüchen definiert werden. Dies geschieht in enger Zusammenarbeit mit dem Auftraggeber. Das Ergebnis ist ein Pflichtenheft, indem alle Fakten des zu entwickelnden Produktes festgehalten werden. Bis zur endgültigen Einführung des Produktes beim Auftraggeber durchläuft das Software-Produkt mehrere Phasen. Diese Phasen werden in diesem Kapitel kurz erläutert.

In Kapitel 5 wird der Begriff Software-Qualitätssicherung definiert. Auch im Software Bereich ist Qualität ein sehr entscheidendes Faktum, denn nicht selten hängt das Leben von einer fehlerfreien Software ab.

Kapitel 6 zeigt das entwickelte Software-Produkt. Es wird anhand von Screenshots⁸ gezeigt, welche Funktionalitäten das Produkt bietet.

In Kapitel 7 beschreibe ich die eingesetzten Hilfsmittel. Diese sind ERwin⁹, PHPed¹⁰ und PHPMyAdmin¹¹.

⁷ HTML = Hypertext Markup Language. Eine einfache Beschreibungssprache für Dokumente, die plattformunabhängig arbeitet

⁸ Abbildung einer Bildschirmanzeige

⁹ Software zum Erstellen von Entity Relationship-Diagrammen

¹⁰ Windows-Editor für verschiedene Programmiersprachen

¹¹ HTML-basiertes Programm zur Verwaltung von mehreren MySQL-Server

1.2 Aufgabenstellung

Bei der Firma AFT soll eine Anwendung für die immer größer werdende Anzahl von Kunden und eingesetzten Lizenzen der einzelnen Soft- und Hardware Produkte entwickelt werden, die die TSG (Technical Support Group) und die einzelnen Entwicklungs-Teams unterstützt und entlastet. Diese Anwendung soll zur Erfassung von Kundenwünschen und AFT-Produktfehlern eingesetzt werden, ebenso zur Verfolgung von Fehlerbeseitigung und Dokumentation.

Bislang erfolgt die Erfassung der Kundenwünsche und AFT-Produktfehler in den einzelnen Entwicklungs-Teams bzw. der TSG größtenteils mit Hilfe von Excel-Tabellen. Da jede Abteilung ihre eigenen Listen führt, ist eine Aus- und Bewertung der eingegangenen Fehlerdaten und Kundenwünsche mit hohem Aufwand verbunden.

Das Datenerfassungssystem besteht aus einem Webserver (Apache) und einem Datenbankserver (MySQL). Die Schnittstelle zwischen den beiden „Servern“ bildet PHP. PHP ist eine serverseitige, in HTML eingebettete Skriptsprache.

Die Anwendung soll von „YaWohl!“, dem Intranet-Angebot der Firma AFT zugänglich sein. Die Gestaltung der Seiten ist in Anlehnung an das bestehende Erscheinungsbild von „YaWohl!“ anzupassen. Durch das Erreichen der Anwendung über das bestehende Intranet-Angebot, ist die Datenerfassung von jedem Arbeitsplatz mit Zugang zum Intranet gewährleistet.

Das Datenerfassungssystem unterscheidet verschiedene Benutzer und ist daher mit einer Benutzerauthentifikation ausgestattet. Somit können sich die Benutzer von jedem Arbeitsplatzrechner mit Zugang zum Intranet in das Datenerfassungssystem anmelden. Nach Anmeldung stehen dem Benutzer die für ihn freigegebenen Funktionen zur Verfügung. Das System soll so „offen“ wie möglich konzipiert sein, wobei eine Anpassung an veränderte Gegebenheiten schnellstmöglich durchführbar sein soll.

1.3 AFT Atlas Fahrzeugtechnik GmbH

Die AFT Atlas Fahrzeugtechnik GmbH wurde 1979 in Werdohl gegründet und ist zu gleichen Teilen ein Tochterunternehmen der LuK Lamellen und Kupplungs-GmbH und der Conti Temic microelectronic GmbH. Sie ist ein auf nationaler wie internationaler Ebene tätiges Entwicklungsunternehmen im Automobilbereich. Neben reinen Auftragsentwicklungen werden auch Eigenentwicklungen im Bereich Kfz-Komponenten und Kfz-Messtechnik durchgeführt. Der Entwicklungsservice erstreckt sich von der Konzepterstellung über die Konstruktion, Teilefertigung, Versuchsdurchführung bis hin zur Kundenunterstützung vor Ort.

AFT entwickelt Soft- und Hardware Produkte für verschiedene Anwendungsfälle ihrer Kunden aus der Automobilindustrie und deren Zulieferbetrieben in eigenen Testständen und Labors. Anwendungsfälle sind z.B. Langzeit-Messungen an Fahrzeugen oder Applikationen für Steuergeräte in der Serien- oder Prototypen-anwendung.

Das Leistungsspektrum der AFT wird durch 5 Fachbereiche abgedeckt.

1) Advanced System Concepts (ASC)

Der Fachbereich ASC entwickelt mechatronische Systeme und Komponenten durch die Kombination der Disziplinen Mechanik, Elektrotechnik und Informatik.

Ein Schwerpunkt ist die Entwicklung und Konstruktion mechanischer Aktuatorik¹². Der andere ist der strategische Entwurf und Programmierung von Funktions-Software (z.B. für Nockenwellenversteller).

Ein weiteres Arbeitsgebiet ist die Simulation von mechanischen Systemen.

¹² Aktuator = Bauelement am Ausgangsteil einer Steuer- oder Regelstrecke, der in Energie oder Massenströme eingreift und darin als veränderlicher Widerstand wirkt.

2) ECU Development and Engineering (ECU)

Der Fachbereich ECU entwickelt zum einen Software für Steuergeräte (wie z.B. Betriebssysteme, Kommunikationshandler, Diagnose- und Funktionssoftware) zum anderen die Hardware für Steuergeräte (vom Schaltungsdesign über den Prototypenaufbau bis hin zur Serie).

Das Spektrum umfasst auch die Lieferung von schlüsselfertigen Hardware-in-the-Loop Systemen und Seriensteuergeräten sowie einen umfassenden Support.

3) ECU Tools (ECT)

Der Fachbereich ECT liefert zum einen automotive¹³ Produkte für die Entwicklung und Validierung mechatronischer Systeme und zum anderen werden kundenspezifische Entwicklungen und Messaufgaben durchgeführt.

Die Produktpalette umfasst:

- Applikationssysteme für Steuergeräte, Drehschwingungsmesstechnik,
- Datenlogger und Vielkanal-CAN-Bus-Temperaturmesstechnik

Zu den Dienstleistungen gehören Fahrzeugausrüstungen, Flottentests, Vor-Ort-Support und Service.

4) Noise, Vibration and Harshness (NVH)

Der Fachbereich Fahrzeugakustik optimiert Komponenten und Gesamtfahrzeuge zur Verbesserung des Geräusch- und Schwingungskomforts. Entwicklungen für Antriebsstrang, Fahrwerk, Karosserie, Hilfsaggregate werden systematisch und kundengerecht durchgeführt.

5) Powertrain Development (PtD)

Der Fachbereich PtD führt auf modernsten Motor- und Getriebeprüfständen Dauerlauf- und Entwicklungsversuche durch.

¹³ Automotiv = selbstfahrend

Das Leistungsspektrum umfasst auch die Entwicklungen für Motor- und Getriebesteuerungen (komplette Systeme und Teilfunktionen für Serie und Prototypen) sowie die Lieferung eines Motorsteuergerätes für Erdgasmotoren.

Die Mitarbeiterzahl der AFT Atlas Fahrzeugtechnik GmbH belief sich im Jahr 2001 auf 140, davon waren 95 Ingenieure, 28 Techniker und 17 Administratoren.

2 Grundlagen

In diesem Kapitel werden die Grundlagen der eingesetzten Hilfsmitteln erläutert. Als erstes wird der Begriff „Intranet“ definiert. Dies ist wichtig, da der AFT-DataLogger in ein bestehendes Intranet eingegliedert wird. Als nächstes folgen die Definitionen für HTML, JavaScript und PHP. Die 3 eingesetzten Techniken bilden die Basis für den AFT-DataLogger. Sie ermöglichen die Datenerfassung sowie die Kommunikation mit dem Datenbanksystem. HTML und JavaScript übernehmen dabei die grafische Darstellung der Anwendung, bzw. die Überprüfung von Benutzereingaben. PHP dient als Schnittstelle zum Datenbanksystem. Als Datenbanksystem wird MySQL zum Einsatz kommen, welches mit PHP zusammenarbeitet, und im Internet bzw. Intranet dank seiner guten Schnelligkeit geschätzt wird.

Die Unterkapitel im einzelnen:

- Intranet
- HTML
- JavaScript
- PHP
- MySQL

2.1 *Intranet*

Die explosionsartige Entwicklung des Internets in den letzten Jahren machten das World Wide Web (WWW) zu einem festen Bestandteil unserer Gesellschaft. Inzwischen sind Millionen von Computern mittels Internet miteinander verbunden und ermöglichen es den Anwendern Zugriff auf nahezu unbegrenzte Sammlungen von Informationen und Ressourcen zu nehmen. Mittlerweile sind viele Firmen im Internet vertreten, um sich dort zu präsentieren, oder auch ihre Produkte online zu verkaufen (E-Commerce). Diese Möglichkeiten sind rund um die Uhr auf dem gesamten Globus abrufbar und wurden somit zu einem wichtigen Marketinginstrument.

Inzwischen gibt es neben dem Begriff „Internet“ noch einen weiteren Begriff, welcher in den letzten Jahren erheblichen Zuspruch gefunden hat, das Intranet. Ein Intranet ist kurz gesagt ein auf Internet-Techniken basierendes System. Als Kommunikations-Protokoll (Backend) wird TCP/IP¹⁴, und als Frontend ein Web-Browser verwendet.

Diese nun eher technische Spezifikation des Intranets, wird nach Höller, Pils, Zlabinger¹⁵ [6] neben dem technischen Gesichtspunkt wie folgt ergänzt:

- **Nach Adressatenkreis**
Eine (virtuelle) Organisation, die einen geschlossenen Benutzerkreis umfasst, aber standortübergreifend ausgelegt sein kann
- **Nach organisatorischen Gesichtspunkten**
Starke formale organisatorische Bindungen und Reglementierungen mit entsprechenden Kontroll- und Sanktionsmechanismen. Es wird versucht, Organisationsmodelle der klassischen DV-Organisation auf Intranet zu übertragen
- **Nach rechtlichen Gesichtspunkten**
Aufgrund arbeits- oder dienstvertraglicher Regelungen relativ strikte Weisungsunterworfenheit der einzelnen Teilnehmer. In Abhängigkeit vom Gebrauch stellen sich unterschiedliche Freiheitsgrade dar. Die konkrete Ausformulierung der Regelungen kann jedoch auch unterbleiben
- **Nach der Aufgabenstellung**
Klassische betriebliche Aufgaben aus den Funktionalbereichen (Logistik, Beschaffung, Rechnungswesen u.a.) dominieren. Immer mehr traditionelle Applikationen sind mit Schnittstellen zur Generierung von HTML-Seiten versehen. Informationsbedarfe können flexibler und zeitnäher abgedeckt werden, wie z.B. Web-Datenbankgateways für Management Informationssysteme, Web-Access für ein Kommunikationstool (Group Wise) oder Datenbestandspflege über HTML-Formulare
- **Nach Freiheitsräumen**
Aufgrund der allgemeinen, genau definierten Reglementierungen nicht oder nur schwer möglich

¹⁴ TCP/IP = **T**ransmission **C**ontrol **P**rotocol / **I**nternet **P**rotocol. Ein Kommunikationsstandard für Netzwerke, auf dem das Internet basiert, der aber auch für lokale Netzwerke verwendet wird

¹⁵ Höller, Pils, Zlabinger, „Internet und Intranet – Auf dem Weg zum Electronic Business“, S.15f.

Die folgende Grafik stellt die einzelnen Elemente eines Intranets anschaulich dar:

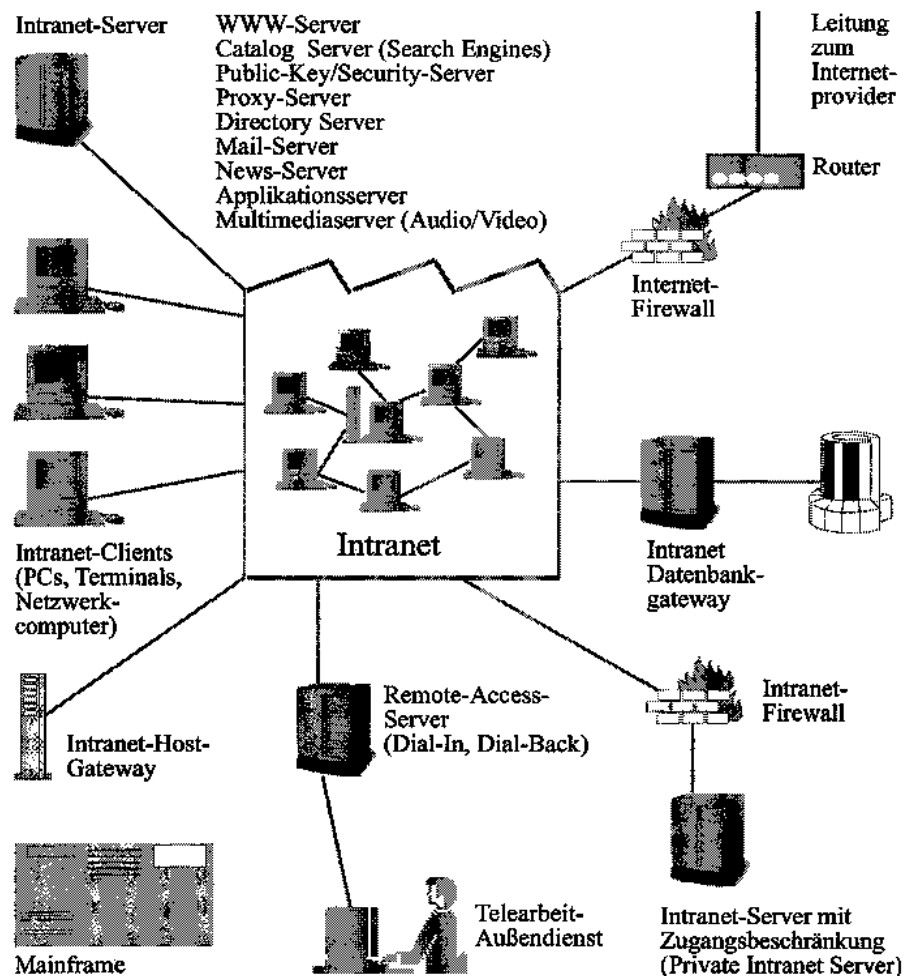


Abbildung 1: Elemente eines Intranets¹⁶

2.2 Hyper Text Markup Language (HTML)

2.2.1 Definition: HTML

Eine Web-Seite unterliegt bestimmten Regeln, damit ein Web-Browser die darin enthaltenen Informationen korrekt darstellen kann. Der Web-Browser sucht nach einer bestimmten Struktur im Dokument, welche durch HTML definiert wird. HTML ist eine spezielle Seitenbeschreibungssprache, also keine Programmier-

¹⁶HÖLLER, PILS, ZLABINGER „Internet und Intranet – Auf dem Weg zum Electronic Business“, S. 157

sprache wie z.B. C++ oder Pascal, sondern sie definiert eine Menge von Regeln mit denen Web-Seiten erstellt werden können.

Entwickelt wurde HTML 1990 von Tim Berners-Lee und Daniel W. Conolly am Europäischen Labor für Teilchenphysik in Genf (CERN). Tim Berners-Lee's Plattform für die Entwicklung von HTML war SGML¹⁷. SGML ist eine ISO¹⁸-Norm, welche 1986 anerkannt wurde und dazu dient, Markierungssprachen wie z.B. HTML formal zu beschreiben. Als Beispiel einer weiteren auf SGML beruhenden Markierungssprache ist hier vor allem XML¹⁹ zu erwähnen. Daniel W. Conolly schrieb seinerseits die HTML-Document Type Definition (DTD), welche eine formale Definition der HTML-Syntax in der Ausdrucksweise von SGML darstellt.

Der explosionsartige Anstieg der Web-Browser und Server seit 1990, sowie der Beginn der starken Kommerzialisierung des World Wide Web haben dazu geführt, dass sich 1994 das CERN²⁰, der MIT²¹ und das INRIA²² zum World Wide Web Consortium, kurz W3C, zusammenschlossen, um die Koordinierung der Standardisierung und Weiterentwicklung von HTML gemeinsam zu steuern. Auf der deutschen Homepage des W3C wird die Aufgabe des W3C folgendermaßen definiert:

„Aufgabe des W3C ist es, dass Web zu seiner vollen Entfaltung zu führen. Dazu werden Technologien entwickelt (Spezifikationen, Richtlinien, Software und Software Tools), die wiederum ein Forum für Informationen, wirtschaftliche Betätigung, Inspirationen, unabhängige Gedanken und kollektives Verständnis schaffen.“²³

Eine HTML-Datei kann mit jedem beliebigen ASCII²⁴-Editor erstellt werden, z.B. mit dem Notepad von Windows. Man benötigt keine Entwicklungsumgebung, um die vielfältigen Möglichkeiten von HTML zu nutzen. Der große Vorteil dabei ist, dass eine einmalig erstellte HTML-Datei auf jedem Betriebssystem lauffähig ist, welches den ASCII-Zeichensatz unterstützt. Somit ist HTML nahezu vollständig

¹⁷ SGML = Standard Generalized Markup Language

¹⁸ ISO = International Organization for Standardization

¹⁹ XML = Extensible Markup Language

²⁰ CERN = Europäischen Labor für Teilchenphysik in Genf

²¹ MIT = Massachusetts Institute of Technology

²² INRIA = National Institute for Research in Computer Science and Control

²³ <http://www.w3.org/Consortium/Offices/Germany/sieben.html>, 04.08.2001

²⁴ ASCII = American Standard Code for Information Interchange

Betriebssystem unabhängig. Diese Tatsache war und ist einer der Gründe für den Erfolg von HTML und somit auch des World Wide Web.

2.2.2 HTML-Versionshistorie

In diesem Unterkapitel wird kurz die Versionshistorie von HTML dargelegt. Die letzte veröffentlichte Version bei Erstellung der Diplomarbeit von HTML ist 4.01. Dies wird auch wahrscheinlich die letzte Version von HTML sein, denn es deutet vieles daraufhin, dass XML der kommende Standard im World Wide Web sein wird.

Version	Jahr	Quelle
HTML 1.0	1992	www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/MarkUp.html
HTML 1.0+	Juli 1993	www.w3.org/MarkUp/HTMLPlus/htmlplus_1.html
HTML 2.0	November 1995	www.w3.org/MarkUp/html-spec/
HTML 3.0	1995	www.w3.org/MarkUp/html3/CoverPage.html
HTML 3.2	Januar 1997	www.w3.org/TR/REC-html32
HTML 4.0	Februar 1998	www.w3.org/TR/1998/REC-html40-19980424
HTML 4.01	Dezember 1999	www.w3.org/TR/1999/REC-html401-19991224

Tabelle 1: HTML-Versionshistorie

2.2.3 Grundstruktur einer HTML-Datei

HTML definiert eine Reihe von Regeln, um Informationen strukturiert darzustellen. Die einzelnen Regeln in HTML werden TAGS²⁵ genannt. Damit ein Web-Browser Informationen von TAGS trennen kann, werden diese in spitzen Klammern geschrieben. Die meisten HTML-TAGS treten paarweise als Start- bzw. End-TAG auf. Der Bereich innerhalb dieses TAG-Paares wird entsprechend dem angewendeten TAG dargestellt. Beispielsweise definiert **Andreas Hurtmann** in einem HTML-Dokument, das mein Name **fett** dargestellt wird.

Eine HTML-Datei besteht grundsätzlich aus drei Teilen:

1. einer Zeile mit der Versions-Information
2. einem beschreibenden Kopfteil
3. dem Dokumentkörper

²⁵ TAGS = kommt vom englischen Wort tag. Tag bedeutet übersetzt *Schild*

Nachfolgend ein HTML-Grundgerüst, das die Minimalanforderungen an ein HTML-Dokument zeigt, welches dem HTML 4.0 Standard genügen soll:

```
1:<!DOCTYPE HTML PUBLIC „-//W3C/DTD HTML 4.0//EN“>
2:<HTML>
3:    <HEAD>
4:        <TITLE> Titel der Seite</TITLE>
5:    </HEAD>
6:    <BODY>
7:        Inhalt der Seite
8:    </BODY>
9:</HTML>
```

- Zeile 1 definiert, nach welcher Spezifikation das HTML-Dokument erstellt wurde
- Zeile 2 und Zeile 9 definieren den Beginn und das Ende eines HTML-Dokumentes
- Zeile 3 definiert den Beginn des Kopfteiles, das in Zeile 5 wieder geschlossen wird. Innerhalb dieses Kopfteiles ist das TAG <TITLE> bzw. </TITLE> vorgeschrieben. Innerhalb dieses <TITLE>-TAG wird der Titel des Dokumentes definiert, das im Browser angezeigt wird. Optional können innerhalb des Kopfteils weitere Dinge definiert werden, so z.B. die Definition einer JavaScript-Funktion, oder die Definition eines Style-Sheets.
- Zwischen dem einleitenden Tag <BODY> in Zeile 6, sowie dem End-Tag </BODY> in Zeile 8 steht die eigentliche anzuzeigenden Informationen. Diese Informationen werden mit den HTML-TAGS formatiert

2.2.4 Definition von Regeln in einem HTML-Dokument

Wie im Punkt 2.3.3 erwähnt, steht der eigentlich darzustellende Text zwischen den HTML-TAGS <BODY> und </BODY>. Neben dem eigentlichen Text stehen dann noch die HTML-spezifische Befehle. Alle HTML-Befehle stehen ebenfalls in TAGS. Die TAGS werden durch spitze Klammern markiert. Fast alle Befehle von HTML bestehen aus einem einleitenden und einem abschließenden Tag. Der

Text dazwischen ist der "Gültigkeitsbereich" für die betreffenden Tags. Ein abschließender TAG unterscheidet sich von einem einleitenden TAG nur durch ein / nach der öffnenden spitzen Klammer.

2.3 JavaScript

2.3.1 Was ist JavaScript?

JavaScript ist kein direkter Bestandteil von HTML, sondern eine eigene Programmiersprache. Diese Sprache wurde zum optimieren von Web-Seiten 1995 von Netscape entwickelt und lizenziert. Dabei hat man sich stark an der Programmiersprache Java von Sun orientiert. Nachdem Netscapes Entwicklung von Sun als "legitimer" Ableger von Java anerkannt wurde, hat man als endgültigen Namen JavaScript gewählt. Auf Basis von JavaScript 1.0 hat Microsoft ebenfalls einen Interpreter in den Internet Explorer integriert. Microsoft bezeichnet seine eigene Entwicklung als JScript. Leider sind die Sprachumfänge nicht vollkommen deckungsgleich. Deshalb beherrschen die jeweils neuen Versionen des Netscape-Browsers auch immer etwas mehr JavaScript als die Konkurrenten, die abwarten, was Netscape an Neuigkeiten implementiert. Mittlerweile gibt es einen Standard, mit dem Namen ECMA²⁶-Script, den Microsoft und Netscape in Zukunft unterstützen wollen.

2.3.2 Übersicht der Browserunterstützung

	<i>Internet Explorer</i>					<i>Netscape Navigation</i>			
	3.x	4.x	5.0	5.5	6.x	2	3.x	4.x	6.x
JavaScript-Version	1.0	1.1	1.2	1.3	1.3	1.0	1.1	1.2	1.3

Tabelle 2: JavaScript Unterstützung der Webbrowser

²⁶ ECMA = European Computer Manufacturers Association

Netscape 2.0 z.B. interpretierte den seinerzeit eingeführten JavaScript-Sprachstandard 1.0, welcher vom Microsoft Internet Explorer erst seit der Version 3.0 verstanden wird. Netscape interpretiert seit Version 3.0 den JavaScript-Standard 1.1. Der Microsoft Internet Explorer interpretiert diesen Sprachumfang von JavaScript erst seit der Produktversion 4.0. Netscape interpretiert seit Version 4.0 den JavaScript-Standard 1.2. Einige Befehle dieser Sprachversion werden auch vom MS Internet Explorer 4.0 interpretiert.

2.3.3 JavaScript ist nicht Java

Vom Namen her scheinen sich Java und JavaScript sehr ähnlich zu sein. Doch unterscheidet sich JavaScript in einigen wesentlichen Punkten von Java. Die nachfolgende Tabelle gibt einen Überblick über die wichtigsten Unterschiede.

JavaScript	Java
<ul style="list-style-type: none"> JavaScript wird direkt in den HTML-Code geschrieben und braucht nicht kompiliert zu werden. Es wird zur Laufzeit vom Browser interpretiert. 	<ul style="list-style-type: none"> Für Java benötigt man das JDK (Java Development Kit) zur Programmierung. Danach wird der Code kompiliert und das Applet liegt als Bytecode in einem sogenannten <i>class</i>-File vor
<ul style="list-style-type: none"> JavaScript ist objekt-basierend und kennt keine Vererbung 	<ul style="list-style-type: none"> Java ist objekt-orientiert, Objekte vererben ihre Eigenschaften
<ul style="list-style-type: none"> JavaScript-Code ist in das HTML-Dokument integriert und kann auf die (meisten) HTML-Elemente zugreifen 	<ul style="list-style-type: none"> Java-Applets sind getrennt von der HTML-Seite.
<ul style="list-style-type: none"> In JavaScript müssen die Typen der verwendeten Variablen nicht deklariert werden. 	<ul style="list-style-type: none"> In Java ist dies zwingend erforderlich
<ul style="list-style-type: none"> In JavaScript werden Objekt-Referenzen erst zur Laufzeit geprüft 	<ul style="list-style-type: none"> in Java bereits während der Kompilierung
<ul style="list-style-type: none"> JavaScript kann ganz einfach analysiert werden. Der JavaScript-Code steht direkt im HTML-Code der Web-Seite. Er steht dort im Klartext 	<ul style="list-style-type: none"> Java-Applets oder Java-Applikationen werden nur mit ihren kompilierten class-Dateien überspielt, und können nicht eingesehen werden

Tabelle 3: Unterschiede Java – Javascript

2.3.4 Wo wird JavaScript definiert?

JavaScript-Programme werden wahlweise direkt in der HTML-Datei oder in separaten Dateien notiert. Sie werden nicht - wie etwa Java-Programme - kompiliert, sondern als Quelltext zur Laufzeit interpretiert. Dazu besitzen die Browser wie Netscape oder Microsoft Internet Explorer entsprechende Interpreter-Software.

In JavaScript gibt es ähnlich wie bei anderen Programmiersprachen Variablen, If-Else-Anweisungen, Schleifen, Funktionen, Methoden, Parameter, Objekte, Eigenschaften und anderes mehr. In JavaScript fehlen aber auch einige Aufgabenbereiche einer "großen" Programmiersprache, z.B. Dinge wie Arbeitsspeicherverwaltung oder Dateioperationen. Außerdem setzt JavaScript auf einer bestimmten Umgebung auf, nämlich auf einer anzuzeigenden oder angezeigten Web-Seite.

Es gibt keine festen Vorschriften dafür, an welcher Stelle einer HTML-Datei ein JavaScript-Bereich definiert werden muss. Es ist üblich, einen solchen Bereich im Kopf der HTML-Datei, also zwischen den TAGS `<head>` und `</head>` zu definieren. Dadurch kann am ehesten gewährleistet werden, dass das JavaScript vom Web-Browser bereits eingelesen ist und zur Verfügung steht, wenn er ausgeführt werden soll.

JavaScript-Code kann automatisch beim Einlesen der HTML-Datei ausgeführt werden. Das ist dann der Fall, wenn JavaScript-Befehle in einem JavaScript-Bereich außerhalb einer selbst definierten Funktion stehen. In solchen Fällen ist es manchmal auch erforderlich, den Script-Bereich innerhalb des HTML-Dateikörpers, also innerhalb von `<body>...</body>`, zu kennzeichnen. Zum Beispiel, wenn man am Ende einer HTML-Seite mit JavaScript dynamisch Datum und Uhrzeit der letzten Änderung schreiben will.

JavaScript kann auch innerhalb von HTML-Tags vorkommen. Das ist dann kein komplexer Programmcode, sondern nur der Aufruf bestimmter Methoden, Funktionen, Objekte oder Eigenschaften. Für den Aufruf gibt es sogenannte Event-Handler. Das sind JavaScript eigene Attribute in HTML-TAGS. Für jeden der möglichen Event-Handler ist festgelegt, in welchen HTML-TAGS er vorkommen darf.

Seit der Version 1.1 von JavaScript (und auch seit HTML 4.0) ist es möglich, den JavaScript-Code in separaten Dateien zu notieren. Das ist sehr nützlich, wenn man gleiche JavaScript-Funktionen in mehreren HTML-Dateien verwenden will. Denn so braucht man den Code nur einmal zu notieren und kann ihn in mehreren HTML-Dateien referenzieren.

2.4 PHP

HTML ermöglicht es dem Entwickler Seiten für das World Wide Web zu programmieren. Ein Nachteil an HTML ist, dass es keine Möglichkeit bietet, dynamische Web-Inhalte zu generieren. Daher ist eine Interaktion mit dem Benutzer mit nur reinem HTML nicht möglich. JavaScript ist eine rein clientseitige Skriptsprache, mit deren Hilfe es möglich ist, einige dynamische Inhalte in die Website zu integrieren. Um aber „echte“ Interaktion mit dem Benutzer realisieren zu können, ist eine Skriptsprache nötig, die auf dem Server ausgeführt wird. Eine dieser Serverseitigen Skriptsprachen ist PHP. PHP generiert dynamisch auf dem Webserver bei Abruf der Seite die Inhalte und schickt diese erst anschließend an den Client zurück. Der User kann an dem Sourcecode nicht erkennen, ob er eine rein statische HTML-Seite übermittelt bekommen hat, oder ob diese bei Abruf generiert wurde. PHP steht für „**P**HP **H**ypertext **P**reprocessor“, dessen Code direkt in eine HTML-Seite hinein geschrieben wird. Der Webserver wertet das Skript bei Abruf der Seite aus und gibt den daraus erstellten reinen HTML-Code an den Client weiter. Der Client-User kann es dem Quelltext nicht entnehmen, dass die Seite dynamisch erstellt wurde. Ein weiterer großer Vorteil von PHP liegt in seiner großen Anzahl von unterstützten Datenbanken. So können HTML-Dateien mit Datenbanken verknüpft werden, was einen erweiterten Funktionsumfang mitliefert. PHP ist nur für den Einsatz im Web-Bereich konzipiert, was ihm einen Vorteil gegenüber anderen Serverseitigen Skriptsprachen verschafft, wie z.B. ASP (Active Server Pages). ASP arbeitet mit VBScript, das nur schwer an die Bedürfnisse eines Webserver angepasst werden kann oder Perl, was vor allem für Anfänger sehr komplex und unstrukturiert aufgebaut ist.

2.4.1 Wie arbeitet PHP?

Die Entstehung einer dynamischen Website mit Hilfe von PHP wird in Abbildung 2 erläutert:

Der Nutzer tippt eine URL²⁷ ein (1), die ein PHP-Skript (2) aufruft. Die im Skript (3) enthaltenen Befehle werden vom PHP-Prozessor (4) erkannt. Wenn Datenbankbefehle enthalten sind, werden diese ausgewertet (5), an die Datenbank gesendet (6) und die Ergebnisse (7) werden in die Seite eingebaut. Die fertige Seite (8) wird dem Webserver übergeben, der sie als reines HTML versendet (10). Der Browser zeigt diese Seite an (11), muss dabei selbst aber von PHP nichts wissen.

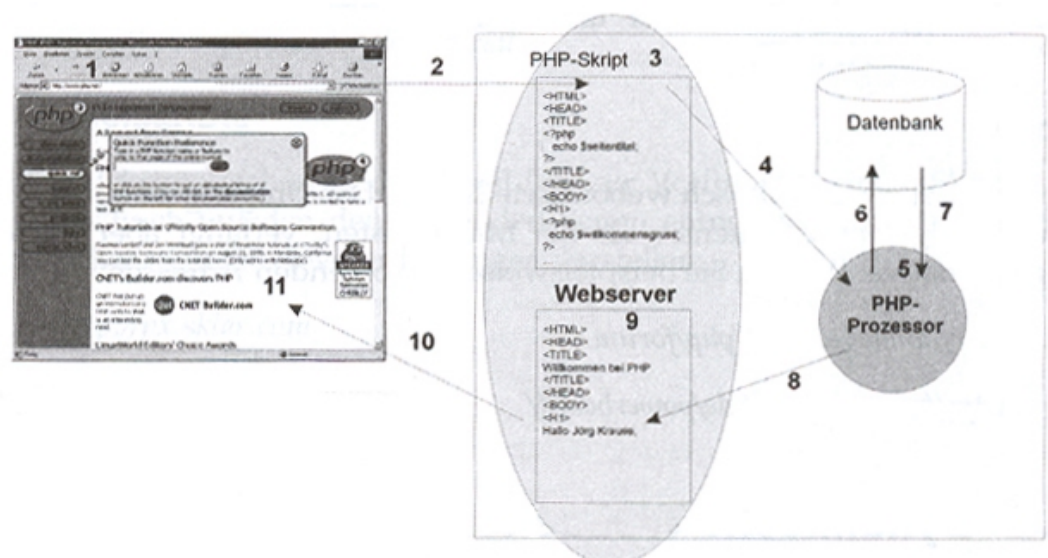


Abbildung 2: So entsteht eine dynamische Website PHP²⁸

2.4.2 Historie von PHP

PHP entstand, als Rasmus Lerdorf einen einfachen Weg suchte, die Besucher seines Online-Lebenslaufs zurückzuverfolgen. Dies war im Jahre 1994 und er

²⁷ URL = **U**niform **R**esource **L**ocator

²⁸ Krause, „PHP4 Webserver-Programmierung unter Windows und Linux“, S. 36

nannte seine kleine Skriptmaschine „Personal Home Page Tools“. Er stellte die 1. Version von PHP ins Netz und ließ somit die freie Verbreitung zu.

1995 wurde dann unter dem Namen PHP/FI die erste wirklich nutzbare Version von PHP veröffentlicht. PHP/FI wird unter den Entwicklern als Version 2 angesehen. Seit dieser 2. Version entwickelte sich PHP rasant weiter. Einen großen Anteil an dieser Entwicklung hatte dabei die freie Verfügbarkeit des Sourcecodes.

Seit 1997 wird PHP von einem Programmiererteam rund um Rasmus Lerdorf weiterentwickelt. 1997 wurde dann auch die 3. Version von PHP veröffentlicht. Diese 3. Version von PHP war ein Durchbruch.

Die zur Zeit aktuelle Version 4 von PHP wurde im Jahr 2000 veröffentlicht. Im Zusammenhang mit der 4. Version von PHP wird häufig der Begriff ZEND verwendet. ZEND²⁹ ist eine Abkürzung für die Entwicklung des neuen PHP-Kernels. ZEND ist ein Kunstwort aus ZEEv und aNDi, den Vornamen der Entwickler Zeev Suraski und Andi Gutman.

In der folgenden Tabelle ist noch einmal die Versionshistorie dargestellt:

<i>Version</i>	<i>Jahr</i>
PHP (Personal Home Page Tools)	1994
PHP/FI (Personal Home Page Tools/Form Interface)	1995
PHP 3.0 (PHP Hypertext Preprocessor)	Juni 1999
PHP4/Zend (PHP Hypertext Preprocessor/ZEND)	2000

Tabelle 4: PHP-Versionen

2.4.3 Wie sieht ein PHP-Skript aus?

PHP steht als Tag innerhalb einer HTML-Datei. Wie bei allen anderen HTML-Tags steht vorne ein Kleiner-als-Zeichen (<) und hinten ein Größer-als-Zeichen (>). Damit der Webserver dieses Tag als PHP-Tag erkennt, steht nach dem Kleiner-als-Zeichen und vor dem Größer-als-Zeichen ein Fragezeichen.

Das folgende Beispiel ist eine kleine HTML-Datei, welche eine PHP-Funktion enthält.

²⁹ ZEND = Kernel der Version 4 von PHP

```
<HTML>
<HEAD>
  <TITLE>Beispiel für PHP-Code innerhalb einer HTML-Datei</TITLE>
</HEAD>
<BODY>
  Das Datum von Heute: <? print(date("l F d, Y")); ?>
</BODY>
</HTML>
```

Abbildung 3 : HTML-Datei mit einer PHP-Funktion vor dem Parsen

Der Webserver erkennt anhand der Dateiendung, dass dieses Dokument durch den PHP-Parser interpretiert werden muss bevor die Seite an den anfordernden Webbrowser geschickt wird. Der PHP-Parser erkennt an den TAGS `<? und ?>` das dazwischen PHP-Code steht. Der Parser wertet die Funktion `date` mit den Formatierungshinweisen `"l F d, Y"` aus. Dabei steht `l` für den aktuellen Wochentag, `F` für den ausgeschriebenen Monatsnamen, `d` für den Tag in Zahlen und `Y` für das Jahr. Wurde die Funktion ausgewertet, wird das Ergebnis mit Hilfe der Funktion `print` in den Sourcecode geschrieben. Ist das Parsen abgeschlossen, wird die nun dynamisch generierte reine HTML-Datei zum aufrufenden Webbrowser geschickt.

```
<HTML>
<HEAD>
  <TITLE>Beispiel für PHP-Code innerhalb einer HTML-Datei</TITLE>
</HEAD>
<BODY>
  Das Datum von Heute: Monday November 12, 2001
</BODY>
</HTML>
```

Abbildung 4: HTML-Datei mit einer PHP Funktion nach dem Parsen

Wird diese Datei im Web-Browser geöffnet, sieht die Ausgabe folgendermaßen aus:

Das Datum von Heute: Monday November 12, 2001

Der Webserver erkennt Dateien, in denen sich PHP-Programmcode befindet, anhand der Dateiendung. Welche Dateiendungen für PHP verwendet wird, kann

in der Konfigurationsdatei des Webserver individuell festgelegt werden. Sehr häufig wird die Endung `.php3` oder `.php` verwendet.

Wie bereits erwähnt wird PHP-Code direkt in die HTML-Datei eingebunden und kann dort an beliebigen Stellen mehrfach durch HTML-Code unterbrochen werden. Wie in allen anderen Programmiersprachen auch, ist es in PHP möglich, den Code mit Kommentaren zu versehen. Kommentare in PHP haben die folgende Form

```
/* dies ist ein Kommentar, auch mehrzeilig */  
// dies ist ein einzeliger Kommentar
```

Datentypen in PHP müssen im Gegensatz zu anderen Programmiersprachen nicht explizit vereinbart werden, eine Typzuweisung wird automatisch vorgenommen. Der Typ einer Variablen wird durch den zugewiesenen Wert bestimmt. PHP kennt folgende Datentypen:

Datentyp	Beschreibung
Integer (int)	Ganzzahl
Double (real)	Fließkommazahl
String	Zeichenkette mit max. 32.768 Zeichen
Array	Datenfeld
Object	Objekt
Boolean	Wahrheitswert (True/False)

Tabelle 5: Datentypen in PHP

Die automatische Typzuweisung folgt dabei folgenden Regeln:

Automatische Typzuweisung

- Eine Zahl ohne Punkt wird als Ganzzahl interpretiert
- Eine Zahl mit Punkt wird als Fließkommazahl interpretiert
- In Anführungszeichen "" eingeschlossene Zeichen werden als String interpretiert
- Bei der Auswertung von Ausdrücken bestimmt der verwendete Operator den Datentyp des Ergebnisses

Grundsätzlich besteht ein PHP-Skript aus einer Anreihung von Anweisungen. PHP stellt verschiedene Strukturierungsmöglichkeiten zur Verfügung, die sich weitgehend an der Sprache C orientieren.

Darunter versteht man:

- Bedingungen
 - **If-else Anweisungen bzw. elseif**
 - **switch**
- Schleifen
 - **while**
 - **do...while**
 - **for**
 - **foreach**

Die foreach Schleife steht seit Version 4 von PHP zur Verfügung.

Mit Hilfe der foreach Schleife kann ein Array bequem durchlaufen werden

- Funktionen und Prozeduren
- OOP (Objektorientierte Programmierung)

Obwohl PHP keine konsequente objektorientierte Sprache ist, stellt PHP Sprachkonstrukte zur Verfügung, mit denen PHP ein wenig Objektorientierung bieten kann. So können in PHP Klassen definiert und vererbt werden.

2.5 MySQL

2.5.1 Definition: MySQL

MySQL ist eine Client/Server Implementierung, die aus einem Server-Dämon „mysqld“ und mehreren Client Programmen besteht.

Es ist ein relationales Datenbank-Management-System, kurz RDBMS. Genauer definiert ist MySQL ein Multi-Threaded, Multi-User RDBMS.

*„**Multi-Threaded** bedeutet, dass jedes Mal, wenn jemand eine Verbindung zum Server herstellt, das Serverprogramm einen Thread oder Prozess erstellt, um die Anfragen des Clients zu bearbeiten. Das erfordert einen extrem schnellen Ser-*

*ver. In der Tat erhält jeder Client, der eine Verbindung zu einem MySQL-Server herstellt, seinen eigenen Thread.*³⁰

Multi-User bedeutet, dass viele Personen die gleiche Applikation zur gleichen Zeit nutzen können.

Eine **relationale Datenbank** ist eine Datenbank, die ihre gesammelten Daten in Tabellen und Spalten organisiert.

MySQL ist aber mehr als eine reine relationale Datenbank, sondern ein **Datenbank Management System** (DBMS). Ein DBMS ist eine Software, die es einem Benutzer ermöglicht, elektronische Datenbanken anzulegen, zu pflegen und zu verwalten. *„Ein DBMS fungiert als ‚Makler‘ (Broker) zwischen der physikalischen Datenbank und den Benutzern dieser Datenbank“*³¹

Die wichtigsten Eigenschaften von MySQL sind Geschwindigkeit, Stabilität und einfache Bedienbarkeit, und wird von allen großen Providern oder auch Suchmaschinenbetreibern eingesetzt.

MySQL ist eine Open-Source-Applikation, welche seit der Version 3.23.19 unter der GPL (General Public License) steht und ist seitdem auch auf Windows-Plattformen lizenzfrei nutzbar.

Genauere Informationen über die Lizenzierung von MySQL erhält man unter folgender Adresse:

<http://www.mysql.com/support/arrangements.html>

³⁰ Maslakowski, „MySQL in 21 Tagen“, S. 32ff.

³¹ Yarger, Reese, King, „MySQL&mSQL“, S. 5ff.

2.5.2 Leistungsmerkmale von MySQL

MySQL ist ANSI³²-SQL92-kompatibel, d.h. MySQL hält sich an alle vom American National Standards Institute festgelegten Regeln.

Stoll und Leiferer³³[17] definieren die Leistungsmerkmale von MySQL folgendermaßen:

- Multi-User-, Multi-Threaded-Architektur.
- Application Program Interface (API) für C, C++, Eiffel, Java, Perl, PHP, Python und TCL.
- Unterstützung von über 20 Betriebssystemplattformen, u.a. Linux, UNIX, FreeBSD, Solaris, SunOS, Windows 9x und NT.
- Quellcode in C und C++. Übersetzung mit vielen Compilern möglich.
- Unterstützte Datentypen: *INTEGER*-Datentypen mit 1, 2, 3, 4 und 8 Byte, *FLOAT*, *DOUBLE*, *CHAR*, *VARCHAR*, *TEXT*, *BLOB*, *DATE*, *TIME*, *DATETIME*, *TIMESTAMP*, *YEAR*, *SET* und *ENUM* Typen.
- Sehr schnelle Implementation von *JOINS*.
- Hochoptimierte SQL-Funktionen.
- Unterstützung von Operatoren und Funktionen im *SELECT*- und *WHERE*-Abschnitt von Abfragen.
- Volle Unterstützung der *GROUP BY* und *ORDER BY* Klauseln und Unterstützung von Aggregatfunktionen.
- Unterstützung von *LEFT OUTER JOIN* gemäß ANSI SQL und ODBC-Syntax.
- Tabellen von verschiedenen Datenbanken können in einer Abfrage verwendet werden.
- Flexibles und sicheres Privileg- und Passwort-System mit Host-basierter Verifikation.
- ODBC-Unterstützung für Windows 9x gemäß ODBC 2.5 Standard.
- Sehr schnelle B-Tree-Tabellen mit Index-Kompression.
- 16 mögliche Indexes pro Tabelle mit maximal 256 Byte Länge.
- Unterstützung von Datensätzen mit fester und variabler Länge.
- Unterstützung von Standardwerten bei Tabellenspalten.

³² ANSI = American National Standards Institute. Eine Standardisierungsorganisation

³³ STOLL, LEIERER, „PHP4 + MySQL“, S. 275f.

- Verwendung von Hash-Tabellen im Arbeitsspeicher als temporäre Tabellen.
- Unterstützung von großen Datenbanken. MySQL-Datenbanken können über 50.000.000 Datensätze enthalten.
- Volle Unterstützung des ISO-8859-1 Latin 1 Zeichensatzes.
- Unterstützung von Alias-Namen für Tabellen und Spalten gemäß SQL92-Standard.
- Die SQL-Befehle *DELETE*, *INSERT*, *REPLACE* und *UPDATE* geben die Anzahl der veränderten Datensätze zurück.
- Die Verbindung zwischen dem MySQL-Datenbankserver und den Clients erfolgt über TCP/IP-Verbindungen bzw. UNIX Sockets oder unter Windows NT über *Named Pipes*.
- Das MySQL-spezifische Kommando *SHOW* kann verwendet werden, um detaillierte Informationen über Datenbanken, Tabellen und Indexe zu erhalten. Mit Hilfe des Befehls *EXPLAIN* können Abfragen optimiert werden.

Unter folgender Adresse kann man sich die Unterschiede zwischen MySQL und anderen Datenbanksysteme grafisch anzeigen lassen. Diese Webseite benutzt die Ergebnisse von *crashme*, einem Programm, das die Funktionalität aller großen relationalen Datenbanken auflistet und vergleicht:

<http://www.mysql.com/information/crash-me.php>

2.5.3 *Einschränkungen bei MySQL*

MySQL hat nicht so eine große Funktionsvielfalt wie andere Datenbanken, z.B. der MS-SQL Server, dennoch enthält es die wichtigsten Features, die eine moderne Relationale Datenbank enthalten sollte. Durch das „weglassen“ von einigen Features ergibt sich auch der Geschwindigkeitsvorteil den MySQL vor anderen Datenbanksystemen hat. Daraus ergibt sich auch die oberste Prämisse der Entwickler von MySQL: Bereitstellen einer größtmöglichen Anzahl von Features, ohne jedoch an Schnelligkeit einzubüßen.

Die folgenden Features sind in der aktuellen Version 3.23.xx noch nicht enthalten, jedoch werden nach und nach einige der Features dazu genommen.

Fremdschlüssel (FOREIGN KEYS)

Fremdschlüssel werden in SQL im Allgemeinen dazu verwendet sicherzustellen, dass die Integrität der verschiedenen Spalten von Tabellen sichergestellt ist. Beispielsweise kann man erzwingen, dass wenn ein Datensatz verändert wird, in einer anderen Tabelle entsprechende Veränderungen automatisch vorgenommen werden.

In absehbarer Zeit wird MySQL um Fremdschlüssel erweitert werden, ein Teil der Syntax ist trotzdem bereits jetzt vorhanden, um die Portierung von Code anderer Datenbanksysteme zu erleichtern. Das Schlüsselwort FOREIGN KEY existiert aus Kompatibilitätsgründen, hat aber keine Funktionalität.

Sichten (Views)

Sichten sind weiter nichts als temporäre Tabellen. In der Sprachreferenz MySQL wird auf die Syntax von temporären Tabellen genau eingegangen. Die Statements werden dadurch zwar etwas komplexer, funktionell können diese aber VIEWS vollständig ersetzen.

Transaktionen

Transaktionen fassen viele Lese und Schreibzugriffe zu einem Prozess zusammen, der dann schnell ausgeführt wird. Hierzu ist ein Transaktionspuffer notwendig, der verzögert alle Änderungen in die Datenbank schreibt. Mit Transaktionen kann man das Schreiben von Clients in eine Datenbank beschleunigen. Der User hat den Eindruck, die Datenbank würde schnell arbeiten. In Wirklichkeit werden diese INSERT und UPDATE Befehle in eine Art QUEUE geschoben, die dann langsam abgearbeitet wird.

MySQL kann jedoch die Befehle, wie INSERT mit einer niedrigen Priorität versehen (LOW_PRIORITY), sodass der Befehl erst dann ausgeführt wird, wenn MySQL eine kurze Pause macht. Dank der Multi-Thread Architektur ist das ohne Probleme möglich.

Eingeschlossene (verschachtelte) SELECT-Abfragen (Subselects)

Die folgende Abfrage ist in MySQL nicht erlaubt:

```
SELECT * FROM table1 WHERE id IN (SELECT id FROM table2)
```

Verschachtelte SELECT-Abfragen kann man durch JOINS recht einfach umschreiben. Es erhöht zwar nicht gerade die Lesbarkeit, ein JOIN ist aber ein vollständiger Ersatz.

Gespeicherte Prozeduren (Stored Procedures)

Gespeicherte Prozeduren sind Makros oder auch Funktionen, bei denen Parameter übergeben werden können. Sie sind sehr praktisch, weil man hiermit auch Personen, die kein SQL beherrschen, ermöglicht, z.B. regelmäßige Statistiken zu erstellen. Der Aufruf einer Zeile im SQL-Client reicht da schon aus.

Trigger

Trigger sind Event-Handler, die bei dem Erreichen von Werten in einer Tabelle gespeicherte Prozeduren oder auch externe Prozesse anstoßen können. MySQL unterstützt keine Trigger. Es ist aber möglich, sich regelmäßig bestimmte Werte in eine ASCII Datei exportieren zu lassen, diese durch ein SHELL-Script abzufragen und dann bestimmte Prozesse oder Abfragen zu starten.

3 Installation und Konfiguration eines LAMP-System

Ein Bestandteil dieser Diplomarbeit ist es, die Installation und Konfiguration eines Webserver mit PHP-Unterstützung durchzuführen. Auf diesem System soll auch der Datenbankserver MySQL eingerichtet werden. In diesem Kapitel werden die einzelnen Installationsschritte aufgezeigt, die nötig sind, um ein LAMP-System einzurichten. LAMP ist eine Abkürzung für die eingesetzte Software, dabei steht **L** für Linux, **A** für Apache, **M** für MySQL und **P** für PHP. LAMP-Systeme sind im Internet-Bereich sehr verbreitete Varianten, häufig kommen aber auch WAMP-Systeme zum Einsatz, die anstelle von Linux, Windows als Betriebssystem verwenden

Die Unterkapitel im einzelnen:

- SuSe Linux 6.3
- Apache 1.3.20
- MySQL 3.23.40
- PHP 4.06

3.1 *SuSe Linux 6.3*

Als Betriebssystem kommt SuSe Linux in der Version 6.3 zum Einsatz. Die SuSE Linux AG in Nürnberg ist ein führender Anbieter von Komplettlösungen auf Basis des Betriebssystems Linux.

Die Suse 6.3 Distribution besteht aus 6 CD's. Auf der ersten CD befindet sich die grafische Installationsroutine Yast2, welche den Benutzer durch die Installation führt. Diese erste CD ist zudem bootfähig, so dass der Rechner direkt von dieser CD booten kann. Sollte der Rechner nicht von einer CD booten können, so liegt eine Boot-Diskette der Distribution bei.

Sobald der Rechner von der CD gebootet hat erscheint der Begrüßungsbildschirm auf dem Monitor und Yast2 wird geladen. Yast2 führt den User Schritt für Schritt durch die Installation.

Als erstes muss die Sprache, das Tastaturlayout sowie die entsprechende Zeitzone ausgewählt werden. Ist dies geschehen wird der Maustyp bestimmt. Nach diesen ersten grundlegend notwendigen Einstellungen hat man nun die Möglichkeit zwischen einer automatischen oder einer interaktiven Installation zu wählen. Die automatische Installation nimmt dem User soweit wie möglich alle Entscheidungen ab und verlangt nur die Eingabe benötigter Daten. Bei der interaktiven Installation wird der User Schritt für Schritt durch die einzelnen Installationspunkte geleitet. Bei der interaktiven Installation folgt nun als erstes die Auswahl der Partition auf der Linux installiert werden soll. Da mein Entwicklungsrechner ein nicht vorinstallierter Rechner war, konnte ich die gesamte Festplatte als Partition anwählen. Im Anschluss daran folgt nun die Auswahl der Software. Da mir zur Entwicklung ein Standard-System genügt, wählte ich dieses als Software-Auswahl, so dass nur die benötigten Softwarekomponenten installiert werden. Als zusätzliche Software ist es aber unbedingt erforderlich, die „Entwicklung“ mit zu installieren, da sich in dieser sämtliche der benötigten Compiler befinden, die man später zum Installieren der weiteren Programme benötigt. Hat man seine Softwareauswahl getroffen folgt nun die Konfiguration des LiLo³⁴-Bootmanagers. Dieser ist nötig, um später Linux überhaupt starten zu können. Die optimale Wahl ist hier, den LiLo im Master Boot Record (MBR) der Primären Partition auf C: zu installieren. Zum Abschluss folgen nun noch die Angaben für den Benutzer des Systems. Der vollständige Name sowie ein Login-Name inkl. Passwort muss festgelegt werden. Zusätzlich muss noch ein Passwort für den User root festgelegt werden. Sind nun alle Angaben vollständig, kann die Installation gestartet werden.

Nach erfolgreicher Installation kann man sich sofort als root ins System einloggen. Jetzt müssen noch einige Konfigurationen von Linux vorgenommen werden, um den Rechner ins bestehende Netzwerk integrieren zu können.

Dazu startet man YaST (Yet another Setup-Tool) am Prompt mit dem Befehl `yast`.

In dem nun erscheinenden Hauptmenu wählt man mit Hilfe der Cursor-Tasten der Tastatur den Punkt „Administration des Systems“.

³⁴ LiLo = **L**inux **L**oader

Als erstes muss die Netzwerkkarte ins System integriert werden. Dazu wählt man den Punkt: „Hardware im System integrieren → Netzwerkkarte konfigurieren“.

Als Netzwerktyp trägt man „eth0“ ein und wählt aus der anschließend erscheinenden Liste seine Netzwerkkarte aus. Ist dies geschehen, bestätigt man und man gelangt wieder zurück zum Punkt „Administration des Systems“.

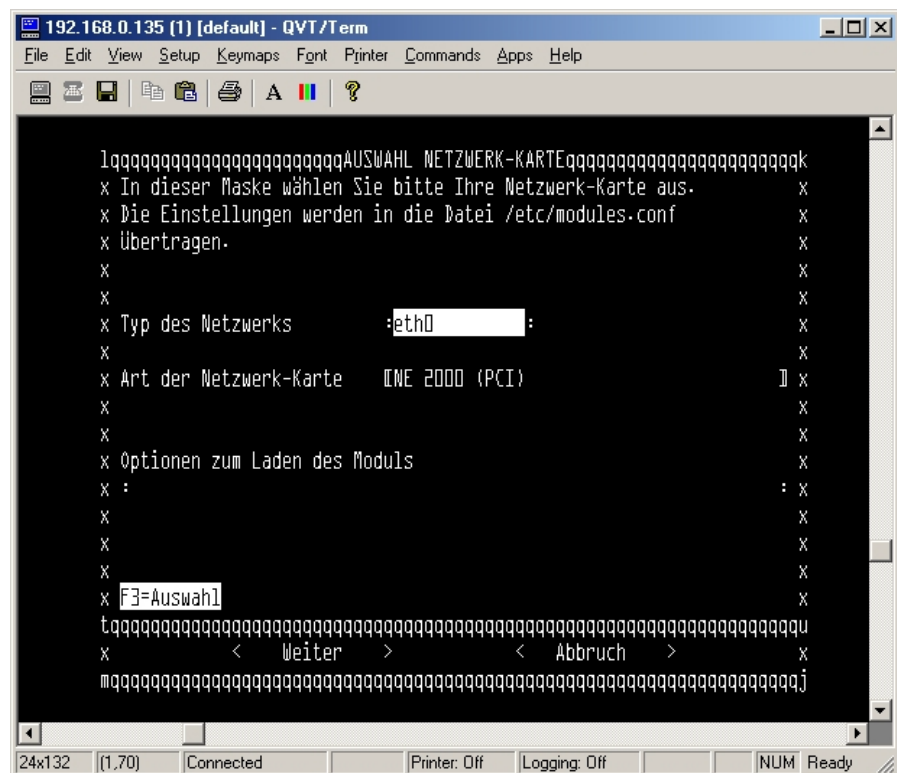


Abbildung 5: Auswahl der Netzwerkkarte

Jetzt muss das Netzwerk konfiguriert werden. Dazu wählt man Netzwerk Grundkonfiguration. Dort wird zunächst ein Device bestimmt. In meinem Falle ist dieses „eth0“, danach die IP-Adresse des Rechners und die Netmask (Standard ist 255.255.255.0). Sind alle Angaben gemacht, kann man das Netzwerk-Device mit F4 aktivieren und die Angaben mit F10 speichern.

```

EINGABE DER NETZWERK-ADRESSEN
In dieser Maske geben Sie die Daten ein, die für die Konfiguration
Ihres Netzwerkes erforderlich sind. Dies sind die IP-Adresse, die
Sie dem Rechner zuweisen wollen (z.B. 192.168.17.42), der gerade
installiert wird und die Netmask Ihres Netzwerkes. Diese ist für
die meisten (kleineren) Netzwerke 255.255.255.0, kann jedoch auch
anders gesetzt sein. Benötigen Sie ein Gateway, um den NFS-Server
erreichen zu können, geben Sie hier die IP-Adresse des Gateway-
Rechners ein.

Typ des Netzwerks:      eth0
IP-Adresse Ihres Rechners: :192.168.0.20      :
Netmask (meist 255.255.255.0): :255.255.255.0      :
Adresse default-Gateway (falls notwendig): :      :
IP-Adresse des PointToPoint-Partners :      :

< Weiter >           < Abbruch >

```

Abbildung 6: Netzwerkgrundkonfiguration

Danach folgt die Änderung des Rechnernamens. Wichtig neben dem Rechnernamen ist die Angabe der Domäne.

```

EINGABE DER NAMEN DES RECHNERS
In dieser Maske wird der Name, unter dem Ihr Rechner im Netz
bekannt ist, angegeben. Der Name besteht aus dem eigentlichen
Rechnernamen und dem Domainnamen. Ein Namensbestandteil darf
Buchstaben, Ziffern und das Zeichen '-' enthalten. Der Domainname
besteht aus mehreren solchen Teilen, die durch Punkte getrennt
sind.

Rechnername :erde      :
Domainname :kosmos.all  :

< Weiter >           < Abbruch >

```

Abbildung 7: Rechnername und Domäne

Zum Abschluss der Konfiguration werden noch die gewünschten Netzwerkdienste konfiguriert. Hervorzuheben ist vor allem der Dienst `inetd` der es ermöglicht, mittels eines Telnet³⁵-Programms auf den Rechner zuzugreifen. Anschließend kann Yast wieder verlassen und das System mit folgendem Befehl neu gebootet werden: `shutdown -r now`. Nach dem Neubooten ist der Rechner im Netzwerk integriert.

³⁵ Telnet = Ein Protokoll, mit dem man auf einen entfernten Computer interaktiv so verbunden werden kann, dass die entfernte Konsole uneingeschränkt zur Verfügung steht

3.2 Apache 1.3.20

Zunächst sollte die aktuelle Version des Apache Webserver aus dem Internet heruntergeladen werden. Downloaden kann man sich den Apache unter folgender URL.

```
http://www.apache.org
```

In dem Download-Bereich findet man neben der aktuellen Version auch viele ältere Versionen des Apache und auch noch im Teststadium befindliche Versionen. Die aktuelle stabile Version ist 1.3.20, welche ich mir im Verzeichnis `/www_sourcen` gesichert habe.

Installation des Apache

Nach dem Download muss als erstes das Archiv entpackt werden

```
$ cd /www_sourcen  
$ gunzip --c apache_1.3.20.tar.gz | tar xvf -
```

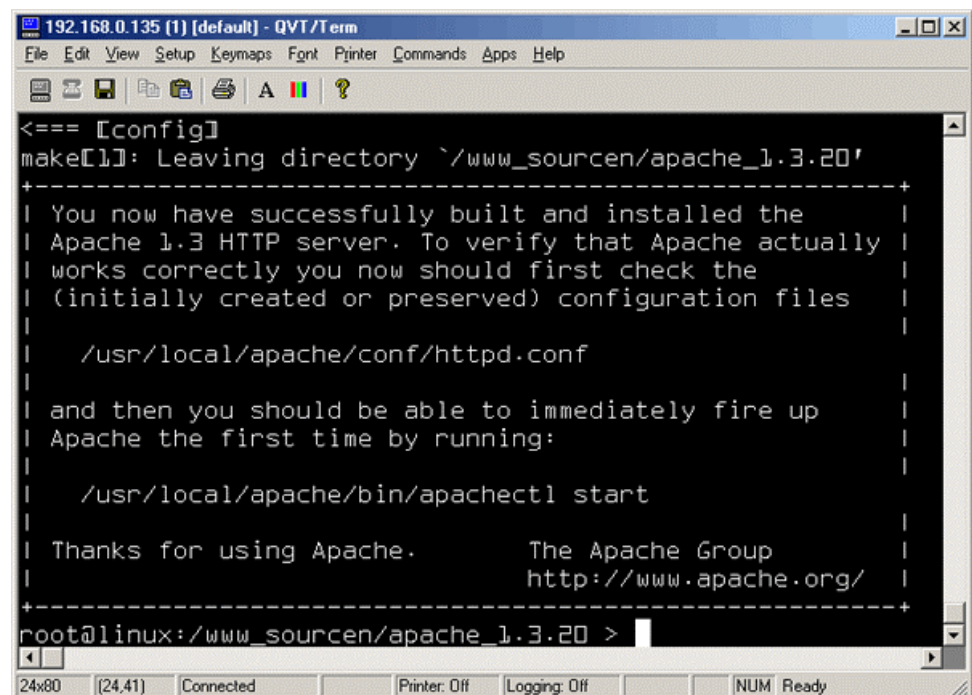
Beim entpacken wird automatisch ein neues Verzeichnis innerhalb des aktuellen Verzeichnisses erstellt. Nach dem Entpacken des Archivs kann nun mit dem Kompilieren des Webserver begonnen werden. Dazu wechselt man in das erstellte Verzeichnis und führt das `configure`-Skript mit den gewünschten Installationsoptionen aus. Standardmäßig genügt es, mit der Option `--prefix` das Installationsverzeichnis festzulegen.

```
$ cd apache_1.3.20  
$ ./configure --prefix=/usr/local/apache
```

Ist das `configure`-Skript abgearbeitet wird die eigentliche Kompilierung mit `make` gestartet und anschließend die endgültige Installation mit `make install` gestartet.

```
$ make  
$ make install
```

Ist `make install` erfolgreich abgearbeitet erhält man folgende Meldung auf dem Bildschirm:



```
192.168.0.135 (1) [default] - QVT/Term
File Edit View Setup Keymaps Font Printer Commands Apps Help

<=== [config]
make[1]: Leaving directory `/www_sourcen/apache_1.3.20'
+-----+
| You now have successfully built and installed the |
| Apache 1.3 HTTP server. To verify that Apache actually |
| works correctly you now should first check the |
| (initially created or preserved) configuration files |
| |
| /usr/local/apache/conf/httpd.conf |
| |
| and then you should be able to immediately fire up |
| Apache the first time by running: |
| |
| /usr/local/apache/bin/apachectl start |
| |
| Thanks for using Apache. The Apache Group |
| http://www.apache.org/ |
+-----+
root@linux:/www_sourcen/apache_1.3.20 >
```

Abbildung 8 : Der Apache wurde erfolgreich installiert!

Konfiguration des Apache

Bevor der Webserver gestartet werden kann muss die Konfigurationsdatei `httpd.conf` editiert werden. Diese befindet sich in `/usr/local/apache/conf`. In dieser Datei müssen zwei Zeilen editiert werden. Zum einen muss der Servername festgelegt werden, und zum anderen das Veröffentlichungsverzeichnis. Der Servername kann frei gewählt werden. Man kann dabei den DNS-Namen eintragen, sofern dieser bekannt ist, oder man trägt dort die IP-Adresse ein. Das Veröffentlichungsverzeichnis kann ebenfalls frei gewählt werden. Es muss nur auf ein real existierendes Verzeichnis zeigen, da sonst der Webserver mit einer Fehlermeldung den Start abbricht.

```
#httpd.conf
ServerName aft_ect_diplom
DocumentRoot "/www"
```

Starten und beenden des Servers

Zum starten und beenden des Servers befindet sich unter `/usr/local/apache/bin` ein Script namens `apachectl`. Mit

`apachectl start` wird der Server gestartet, und mit `apachectl stop` gestoppt

Automatisches Starten von Apache beim Hochfahren des Rechners und automatisches Beenden beim Herunterfahren

Das Script `apachectl` wird dazu ins Verzeichnis `/etc/rc.d` kopiert. Innerhalb dieses Verzeichnisses existieren für jedes verschiedene Systemlevel ein Unterverzeichnis. In unserem Falle ist das Verzeichnis `rc0.d` und `rc2.d` interessant. `rc0.d` ist das Shutdown-Level, d.h. wenn Linux herunter gefahren wird, werden sämtliche Skripte in diesem Verzeichnis ausgeführt. `rc2.d` ist das Start-Level für den normalen ASCII-Text Modus, d.h. beim Hochfahren werden in diesem Verzeichnis sämtliche Skripte ausgeführt. Damit das `apachectl`-Skript also von diesen beiden Leveln ausgeführt wird, werden in diesen Verzeichnisse Symbolische Links auf `apachectl` erstellt.

```
$ ln -s ../apachectl S99apache
```

3.3 MySQL 3.23.40

Bevor mit der Installation von MySQL begonnen werden kann, sind zwei Dinge im vorhinein zu klären.

1. Wird MySQL aus dem Quellcode oder aus einer Binary installiert?

Binaries sind bereits für bestimmte Betriebssysteme vorkompilierte MySQL-Versionen. Binary Versionen sparen Zeit bei der Installation, allerdings ist die individuelle Anpassbarkeit beschränkt. Binaries setzen außerdem voraus, dass Sie als „root“ installiert werden, da in den Binaries die Pfadinformationen fest kodiert sind, und diese Pfadanlegungen in bestimmten Verzeichnissen „root“-Privilegien voraussetzen.

Eine Installation über den Quellcode erfordert einen C-Compiler und andere Entwicklungswerkzeuge. Eine SuSe-Linux Standardinstallation mit Entwicklungswerkzeugen beinhaltet alle benötigten Programme für die Installation.

2. Unter welchem Benutzer wird MySQL installiert?

Um MySQL zu verwenden sind keine „root“³⁶-Rechte nötig. Falls man keine „root“-Rechte besitzt muss man MySQL in seinem Home-Verzeichnis installieren. Wie bereits vorhin erwähnt ist dann allerdings nur eine Installation über den Quellcode möglich, da in den Binaries feste Pfadangaben enthalten sind.

Im Falle dieser Diplomarbeit fiel die Entscheidung auf eine Installation als „root“ aus einer Quellcode-Version.

Als erstes folgt nun der Download der benötigten MySQL Version. Als Download-Quellen für MySQL Quellcode und Binaries stehen u.a. folgende Internet-Adressen in Deutschland zur Verfügung:

	URL
Uni Bonn	http://www.wipol.uni-bonn.de/MySQL/ ftp://ftp.wipol.uni-bonn.de/pub/mirror/MySQL
Wolfenbüttel	http://www.fh-wolfenbuettel.de/ftp/pub/database/mysql/ ftp://ftp.fh-wolfenbuettel.de/pub/database/mysql/
Staufen	http://mysql.staufen.de

Tabelle 6 : URL's zum Downloaden von MySQL

Anhand der Dateinamen lässt sich erkennen, ob eine MySQL-Datei

- eine Quellcode oder Binary ist, im Falle eines Binary auch für welches System und
- ob es eine stabile oder eine noch in der Entwicklung befindliche Version ist

³⁶ Bezeichnung für einen speziellen Benutzer eines Systems. Dieser Benutzer hat üblicherweise keinerlei Einschränkungen im System

Als Beispiel:

mysql-3.22.19-beta-sgi-irix6,4-mip.tgz

⇒ Unstabile Binary

mysql-3.22.31-pc-linux-gnu-i586.tgz

⇒ Stabiles Binary für Linux-Systeme

mysql-3.23.40.tar.gz

⇒ Stabiler Quellcode

Die zur Zeit stabile Version ist die 3.22.40, welche ich mir im Verzeichnis /www_sourcen gesichert habe.

Installation der Quelldistribution

Als erstes muss das Archiv wie folgt entpackt werden

```
$ cd /www_sourcen  
$ gunzip -c mysql-3.23.40.tar.gz | tar xvf -
```

Beim Entpacken wird automatisch ein Verzeichnis innerhalb des aktuellen Verzeichnisses erstellt. Nachdem nun das Archiv entpackt wurde können die Source-Dateien kompiliert werden. Dazu wird zunächst das `configure`-Skript gestartet.

```
$ cd mysql-3.23.40  
$ ./configure
```

Um das `configure`-Skript im aktuellen Verzeichnis auszuführen wird dem Befehl den Verweis auf das aktuelle Verzeichnis vorangestellt (`./`). Das `configure`-Skript kann auch mit verschiedenen Zusatzoptionen gestartet werden, welche aber für die Diplomarbeit nicht relevant erschienen.

Ist das `configure`-Skript abgearbeitet wird die eigentliche Kompilierung mit `make` gestartet. Nachdem die Kompilierung abgeschlossen ist, installiert der Befehl `make install` alle Dateien in die dafür vorgesehenen Verzeichnisse.

```
$ make  
$ make install
```

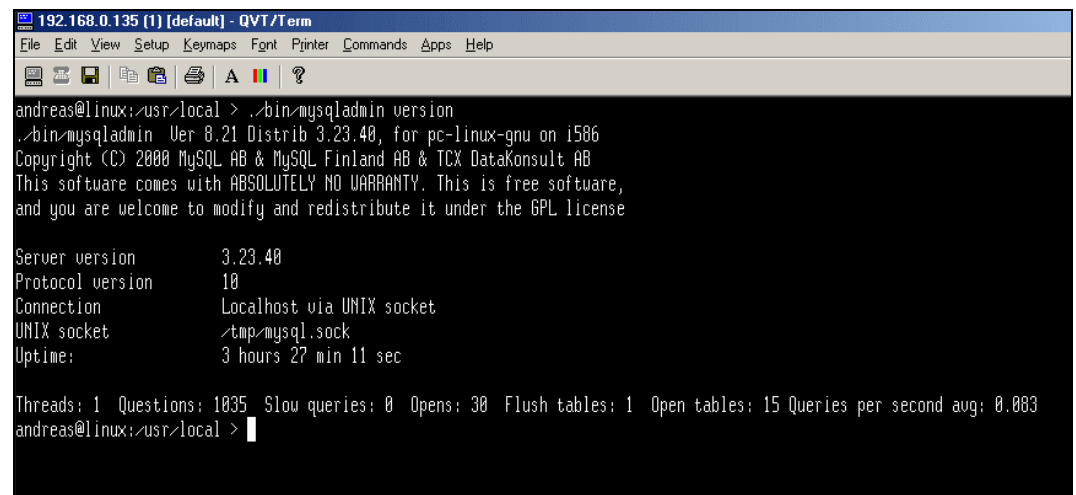

Damit ist die Grundinstallation von MySQL abgeschlossen. Um MySQL aber starten zu können, benötigt man eine Grund-Datenbank-Struktur, die u.a. die administrativen Tabellen enthalten. Um diese zu erzeugen ist folgender Befehl einzugeben:

```
$ ./scripts/mysql_install_db
```

Mit diesem Befehl wird gleichzeitig auch der Server-Dämon gestartet. Um zu testen, ob der MySQL-Server auch wirklich läuft wechselt man in das Installationsverzeichnis, welches sich standardmäßig in `/usr/local` befindet. In diesem Verzeichnis gibt man folgenden Befehl ein:

```
$ cd /usr/local  
$ ./bin/mysqladmin version
```

Wenn der Server ordnungsmäßig läuft erhält man folgende Bildschirmausgabe:



```
192.168.0.135 [1] [default] - QVT/Term  
File Edit View Setup Keymaps Font Printer Commands Apps Help  
andreas@linux:/usr/local > ./bin/mysqladmin version  
./bin/mysqladmin Ver 8.21 Distrib 3.23.40, for pc-linux-gnu on i586  
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX DataKonsult AB  
This software comes with ABSOLUTELY NO WARRANTY. This is free software,  
and you are welcome to modify and redistribute it under the GPL license  
  
Server version          3.23.40  
Protocol version        10  
Connection              Localhost via UNIX socket  
UNIX socket             /tmp/mysql.sock  
Uptime:                 3 hours 27 min 11 sec  
  
Threads: 1 Questions: 1035 Slow queries: 0 Opens: 30 Flush tables: 1 Open tables: 15 Queries per second avg: 0.003  
andreas@linux:/usr/local > |
```

Abbildung 9: Der MySQL-Server läuft

Automatisches Starten und Beenden des MySQL-Dämons beim Hochfahren und Beenden des Betriebssystems

Die Distribution von MySQL enthält im Verzeichnis `support_files` eine Datei namens `mysql.server`. Dieses Unix-Shellskript dient als Startup/Shutdown-Skript. Es wird ins Verzeichnis `/etc/rc.d` kopiert. Innerhalb dieses Verzeichnisses existieren für jedes verschiedene Systemlevel ein Unterverzeichnis. In unserem Falle ist das Verzeichnis `rc0.d` und `rc2.d` interessant. `rc0.d` ist das

Shutdown-Level, d.h. wenn Linux herunter gefahren wird, werden sämtliche Skripte in diesem Verzeichnis ausgeführt. `rc2.d` ist das Start-Level für den normalen ASCII-Text Modus, d.h. beim Hochfahren werden in diesem Verzeichnis sämtliche Skripte ausgeführt. Damit das `mysql.server` Skript also von diesen beiden Leveln ausgeführt wird, werden in diesen Verzeichnisse Symbolische Links auf `mysql.server` erstellt.

Grundkonfiguration

Der erste wichtige Konfigurationsschritt ist, ein Passwort für den Datenbank-User `root` einzurichten. Dazu führt man folgenden Befehl im Installationsverzeichnis aus.

```
$ ./bin/mysqladmin -u root password 'neuespassword'
```

Somit kann nur derjenige, der das `root`-Passwort kennt, Änderungen an den administrativen Tabellen vornehmen.

3.4 PHP 4.06

Zunächst muss das Archiv der aktuellen Version aus dem Internet heruntergeladen werden. Die zur Zeit aktuelle Version von PHP ist 4.06 und heruntergeladen kann man diese Version unter folgender Adresse:

```
http://www.php.net
```

Installation

Nach dem Download muss zunächst das Archiv entpackt werden.

```
$ cd /www_sourcen  
$ gunzip -c php-4.06.tar.gz | tar xvf -
```

Beim Entpacken wird automatisch ein Verzeichnis innerhalb des aktuellen Verzeichnisses erstellt. Nach dem das Archiv entpackt wurde können die Source-Dateien kompiliert werden. Dazu startet man zunächst das `configure`-Skript

mit den gewünschten Installationsoptionen. Zu beachten ist, dass bevor das `configure`-Skript gestartet wird, der Apache wie in 3.2 beschrieben kompiliert wurde, sonst meldet PHP einen Fehler. Die folgende `configure`-Anweisung reicht für die meisten Standardanwendungen unter PHP aus.

```
$ cd php-4.06
$ ./configure --with-apache=../apache_1.3.20/ --enable-track-vars
```

Nach Abarbeitung des `configure`-Skripts wird die eigentliche Kompilierung mit `make` gestartet und die endgültige Installation mit `make install` abgeschlossen.

```
$ make
$ make install
```

Anschließend muss zur Einbindung des PHP-Moduls der Apache Webserver neu kompiliert werden. Dazu wechselt man wieder ins Installationsverzeichnis und ruft dort wieder das `configure`-Skript auf. Als zusätzliche Installationsoption wird nun das PHP-Modul aktiviert. Danach wird die Kompilierung wieder gestartet.

```
$ cd ..
$ cd apache_1.3.20
$ ./configure --prefix=/usr/local/apache --activate-module=src/modules/php4/libphp4.a
$ make
$ make install
```

Nach Abschluss der Neukompilierung muss dem Apache Webserver in seiner Konfigurationsdatei mitgeteilt werden, dass er Dateien mit der Endung `.php` als PHP-Programme erkennt, und diese entsprechend mit dem PHP-Parser verarbeitet. Dazu muss folgende Zeile in der Konfigurationsdatei eingefügt werden.

```
AddType application/x-httpd-php .php
```

Zum Abschluss der Installation wird die PHP Initialisierungsdatei nach `/usr/local/lib` kopiert.

```
cp php.ini-dist /usr/local/lib/php.ini
```

Die Initialisierungsdatei bietet die Möglichkeit, PHP an die Bedürfnisse einer Web-Anwendung anzupassen. Die Grundeinstellungen der Initialisierungsdatei erfüllen bereits die Anforderungen der meisten Web-Anwendungen.

4 Software Entwicklung

Eine der Aufgaben der Software-Entwicklung ist es, aus einem geplanten Produkt ein fertiges Produkt zu entwickeln, welches die geforderten Qualitätseigenschaften besitzt. Der Prozess einer Software-Entwicklung wird in eine Reihe von verschiedenen Aktivitäten aufgeteilt, deren einzelne Ergebnisse Teilprodukte sind, welche nach zeitlichen, begrifflichen, technischen oder organisatorischen Kriterien zu Phasen zusammengefasst werden. Anhand der sechs Phasen-Gliederung von Balzert möchte ich diese kurz vorstellen.

Die Unterkapitel im einzelnen:

- Die Planungsphase – Phase 1
- Die Definitionsphase – Phase 2
- Die Entwurfsphase – Phase 3
- Die Implementierungsphase – Phase 4
- Die Abnahme- und Einführungsphase – Phase 5
- Die Wartungs- und Pflegephase – Phase 6

4.1 Die Planungsphase – Phase 1

Bevor mit dem Entwickeln einer Software begonnen werden kann, muss festgestellt werden, ob das zu entwickelnde Produkt überhaupt durchführbar ist. Dazu wird das zu entwickelnde Programm aus verschiedenen Sichten betrachtet. Die einzelnen Sichten sind:

- *fachliche Sicht:*

Die Ergebnisse einer fachlichen Sicht münden in einem Lastenheft. Das Lastenheft wird meist so aufbereitet, das es später in der Definitionsphase erweiterbar ist, und so zu einem Pflichtenheft ausgebaut wird. Ein Lastenheft hat, wie bereits kurz angedeutet die Aufgabe, sämtliche fachliche Basisanforderungen in einem gegliederten Dokument festzuhalten. Zu beachten ist vor allem, dass in einem Lastenheft nur das „Was“ und nicht das „Wie“ beschrieben ist, und sich der Inhalt bewusst auf die fundamentalen Eigenschaften des Produktes

konzentriert. Folgende Abbildung zeigt ein Beispiel für eine Gliederung eines Lastenheftes:

1. Zielbestimmung
2. Produkteinsatz
3. Produktfunktionen
4. Produktdaten
5. Produktleistungen
6. Qualitätsanforderungen
7. Ergänzungen

Tabelle 7 : Beispiel für eine Gliederung eines Lastenheftes

- Ökonomische Sicht

Für eine Beurteilung der ökonomischen Sicht ist es nötig festzustellen, wie groß der Zeitaufwand des zu entwickelnden Produktes ist. Zur Ermittlung des Zeitaufwandes kommt eine Schätzmethode zum Einsatz. Die meisten Schätzmethoden basieren dabei auf das Schätzen des Umfanges des Software-Produktes in Anzahl der Programmzeilen, auch bezeichnet als LOC (Lines of Code)

Eine Aufwandsschätzung sollte nach folgenden Basismethoden erfolgen, die sinnvollerweise kombiniert eingesetzt werden.

- Analogiemethode
- Relationsmethode
- Multiplikatormethode
- Gewichtungsmethode
- Methode der parametrischen Gleichungen
- Prozentsatzmethode

- personelle Sicht

„Eine durchschnittliche Software-Entwicklung liefert ungefähr 350 Quellcodezeilen (ohne Kommentare) pro Ingenieurmonat. Dabei umfasst die Ingenieurzeit alle Phasen von der Definition bis zur Implementierung“

Ausgehend von dieser Zahl kann mit Zuhilfenahme der Ergebnisse aus ökonomischer Sicht ein geschätzter Aufwand in Mitarbeiterjahren (MJ) oder Mitarbeitermonaten (MM) bzw. Personenmonaten (PM) er-

rechnet werden. Dazu werden die geschätzten LOC aus der ökonomischen Sicht durch die Anzahl der Quellcodezeilen dividiert.

Alle Ergebnisse der Planungsphase münden in einer Durchführbarkeitsstudie (feasibility study). Die Durchführbarkeitsstudie dient zur Entscheidung über die weitere Vorgehensweise: stop or go.

4.2 Die Definitionsphase – Phase 2

In der Definitionsphase werden die Produktanforderungen definiert.

„Anforderungen (requirements) legen die qualitativen und quantitativen Eigenschaften eines Produkts aus der Sicht des Auftraggebers fest. Die systematische Vorgehensweise, um die Anforderungen in einem iterativen Prozess zu ermitteln bezeichnet man als Systemanalyse (requirements engineering)

Der Iterative Prozess beinhaltet folgende Tätigkeiten:

Anforderungen:

- Ermitteln*
- Festlegen und beschreiben*
- Analysieren*
- u.U. animieren, simulieren und ausführen*
- verabschieden“³⁷*

Die Definitionsphase erstellt aus den zuerst sehr vagen, verschwommenen, unzusammenhängenden, unvollständigen und widersprüchlichen Anforderungen ein vollständiges, konsistentes und eindeutiges Anforderungsdokument. Die Produktdefinition dient als Basis für die Abnahme des fertigen Produktes und stellt somit ein wichtiges Dokument in der Software-Entwicklung dar.

Das Ergebnisdokument der Anforderungsdefinition wird als Pflichtenheft bezeichnet. Der Inhalt des Pflichtenheftes besteht aus dem fachlichen Funktions-, Daten-, Leistungs- und Qualitätsumfang des Produktes. Das Pflichtenheft sollte so abgefasst sein, dass es als Basis eines Vertrages dienen kann. Anhand des

³⁷ Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 92

Pflichtenheftes soll das fertige Produkt abgenommen werden können. Balzert³⁸ [2] baut das Pflichtenheft folgendermaßen auf.

1. „Zielbestimmung

1.1 Musskriterien

Hier wird aufgeführt, welche Leistungen für das Produkt unabdingbar sind, damit es für den vorgesehenen Einsatzzweck verwendet werden kann. Sie müssen auf jedenfall erfüllt werden.

1.2 Wunschkriterien

Wunschkriterien beschreiben Wünsche an das zu entwickelnde Produkt, die nicht unabdingbar sind, deren Erfüllung aber so gut wie möglich angestrebt werden sollte.

1.3 Abgrenzungskriterien

Abgrenzungskriterien sollen deutlich machen, welche Ziele mit dem Produkt bewusst nicht erreicht werden sollen. Da die Wünsche an ein Produkt im allgemeinen sehr umfangreich und oft leicht formulierbar sind, soll dieser Abschnitt dazu dienen, Abgrenzungen des Produktes zu definieren.

2. Produkt-Einsatz

2.1 Anwendungsbereiche

2.2 Zielgruppen

2.3 Betriebsbedingungen

Da der geplante Produkteinsatz wesentliche Auswirkungen auf die funktionale Mächtigkeit und auf die Qualitätsmerkmale hat, werden in diesem Abschnitt die Anwendungsbereiche, z.B. Textverarbeitung im Büro, und die Zielgruppen, z.B. Sekretärinnen, Schreibkräfte, definiert. Unter Umständen sollte auch festgelegt werden, von welchen Voraussetzungen, z.B. bezüglich des Qualifikationsniveaus des Benutzers, ausgegangen wird. Ebenfalls kann es sinnvoll sein, explizit anzugeben, für welche Anwendungsbereiche und Zielgruppen das Produkt nicht vorgesehen ist, z.B. für den DV-unkundigen Benutzer.

Deckt das Produkt verschiedene Anwendungsbereiche und Zielgruppen ab, dann ist eine Aufführung der unterschiedlichen Bedürfnisse und Anforde-

³⁸ Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 106ff.

runge n nötig. Unter Betriebsbedingungen werden folgende Punkte beschrieben:

- *physikalische Umgebung des Systems,*
- *tägliche Betriebszeit,*
- *ständige Beobachtung des Systems durch Bediener oder unbeaufsichtigter Betrieb.*

3. Produkt-Umgebung

3.1 Software

Unter Software wird angegeben, welche Software-Systeme (Betriebssystem, Laufzeitsystem, Datenbank, Fenstersystem usw.) auf der Zielmaschine (Maschine, auf der das fertiggestellte Produkt eingesetzt werden soll) zur Verfügung stehen.

3.2 Hardware

Unter Hardware wird aufgeführt, welche Hardware-Komponenten (CPU, Peripherie, z.B. Grafikbildschirm, Drucker) in minimaler und maximaler Konfiguration für den Produkteinsatz vorgesehen sind.

3.3 Orgware

Unter Orgware wird aufgeführt, unter welchen organisatorischen Randbedingungen bzw. Voraussetzungen das Produkt eingesetzt werden soll.

3.4 Produkt-Schnittstellen

Unter Produkt-Schnittstellen wird das Produkt in eine bestehende oder geplante Produkt-Familie eingeordnet oder die geforderten bzw. genutzten Schnittstellen zu anderen Produkten werden definiert bzw. vereinbart (z.B. Schnittstelle zum Ferndiagnosesystem). Außerdem kann auf andere Produkte verwiesen werden, die denselber Anwendungsbereich abdecken oder dieselbe Zielgruppe ansprechen.

4. Produkt-Funktionen

4.1 Funktion 1

4.2 Funktion 2 usw.

Unter Produkt-Funktionen erfolgt die funktionale Beschreibung des Produktes aus Benutzersicht. Dieses Kapitel sollte in so viele Abschnitte gegliedert werden, wie das Produkt Funktionen oder Funktionsbereiche aufweist.

5. Produkt-Daten

5.1 Daten 1

5.2 Daten 2 usw.

Beschreibung der langfristig zu speichernden Daten aus Benutzersicht. Referenzierung: /D10/ usw.

6. Produkt-Leistungen

Unter Produkt-Leistungen werden alle Anforderungen aufgeführt, die zeitbezogen oder umfangsbezogen sind. Die einzelnen Leistungsanforderungen werden analog wie die Funktionsanforderungen nummeriert, allerdings mit dem vorangestellten Buchstaben L.

7. Benutzungsoberfläche

In diesem Kapitel werden grundlegende Anforderungen an die Benutzungsoberfläche festgelegt. In Abhängigkeit vom Produkt sollten folgende Gesichtspunkte berücksichtigt bzw. festgelegt werden:

- *Bildschirmlayout*
- *Drucklayout*
- *Tastaturbelegung*
- *Dialogstruktur usw.*

Die Festlegungen sollten sich auf die produktspezifischen Ausprägungen beschränken, die nicht durch das Qualitätsmerkmal Benutzungsfreundlichkeit im Kapitel 8 Qualitäts-Zielbestimmung abgedeckt wird. Die einzelnen Anforderungen werden analog wie die Funktionsanforderungen nummeriert, allerdings mit dem vorangesetzten Buchstaben B. Bei Produkten, die keine Benutzungsoberfläche besitzen, werden hier analog die Schnittstellenkonventionen beschrieben, die für das anwendende System wichtig sind.

8. Qualitäts-Zielbestimmung

In diesem Kapitel wird festgelegt, welche Qualitäts-Merkmale das zu entwickelnde Produkt in welcher Qualitätsstufe besitzen soll. Voraussetzung für

die Qualitäts-Zielbestimmung ist, dass die Qualitäts-Merkmale in operationalisierter Form vorliegen. Die operationalisierten Qualitäts-Merkmale sind als Anhang dem Pflichtenheft beizufügen, wenn sie nicht als allgemeine Richtlinie (Standard, Werknorm) zur Verfügung stehen.

9. Globale Testszenarien/Testfälle

9.1 Testfall 1

9.2 Testfall 2 usw.

In diesem Kapitel werden anwendungsbezogene Testfälle zusammengestellt, die im allgemeinen mehrere Produkt-Funktionen in Anspruch nehmen. Während die Testfälle pro Funktion aus den Funktionsanforderungen abgeleitet werden, sollten in diesem Kapitel globale Testfälle aufgeführt werden. Diese Testfälle sind dann für den Abnahmetest zu verwenden.

10. Entwicklungs-Umgebung

In diesem Kapitel wird die Entwicklungs-Umgebung des Produktes beschrieben. Es wird festgelegt, welche Konfiguration bzgl. Software, Hardware und Orgware für die Entwicklung des Produktes benötigt wird. Diese Festlegungen sind insbesondere dann notwendig, wenn Entwicklungs- und Zielmaschine unterschiedlich sind.

Bei Entwicklungs-Schnittstellen ist unter Umständen aufzuführen, über welche einzuhaltenden Hardware- und Software-Schnittstellen Entwicklungs- und Zielrechner gekoppelt sind. Unter Software ist insbesondere aufzuführen, welche Software-Werkzeuge, z.B. Compiler usw., benötigt werden.

11. Ergänzungen

In diesem Kapitel werden Ergänzungen oder spezielle Anforderungen beschrieben, die über die aufgeführten Kapitel 1 bis 10 hinausgehen. Beispielsweise können hier Installationsbedingungen festgelegt werden wie:

- bauliche und räumliche Voraussetzungen,*
- Bereitstellung von Testdaten,*
- Bereitstellung von Hilfspersonal*

Außerdem können hier zu berücksichtigende Normen, Vorschriften, Patente und Lizenzen aufgeführt werden.“

In den folgenden Unterkapiteln werden die Basiskonzepte einer Definitionsphase beschrieben, die im Pflichtenheft des AFT-DataLogger verwendet wurden. Darunter zählen der Funktionsbaum, das Datenflussdiagramm und das ER-Diagramm³⁹.

4.2.1 Funktionsbaum

Bei einem Funktionsbaum werden für eine gegebene Problemstellung Funktionen hierarchisch gegliedert.

Eine Funktion beschreibt eine Tätigkeit oder eine klar umrissene Aufgabe innerhalb eines größeren Zusammenhang. Ein Funktionsbaum entsteht, wenn allgemeine Funktionen in spezielle Teilfunktionen gegliedert werden. Grafisch wird der Funktionsbaum in Form eines Baumdiagramms dargestellt.

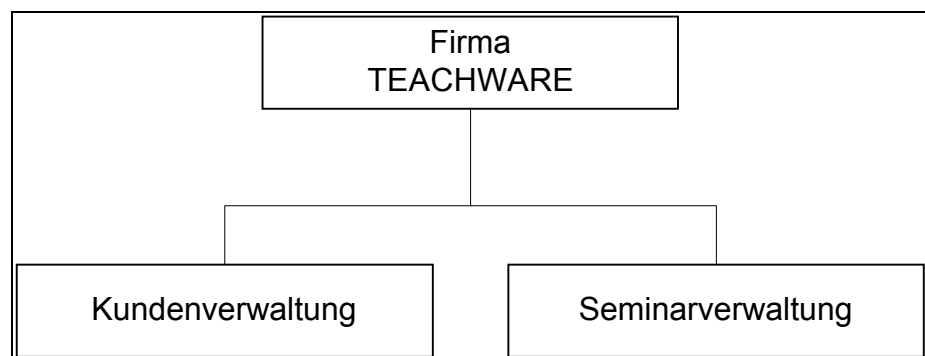


Abbildung 10: Beispiel eines Funktionsbaumes

Bei der Erstellung von Funktionsbäumen sind zwei Regeln hervorzuheben:

1. Unter einer gemeinsamen Vaterfunktion sollen nur Funktionen angeordnet sein, die fachlich eng zusammengehörende Tätigkeiten beschreiben. Was eng zusammengehört, kann nur mit Fachwissen entschieden werden.
2. Auf einer Hierarchieebene sollen Funktionen angeordnet sein, die sich auf gleichem Abstraktionsniveau befinden.

³⁹ ER-Diagramm = Entity Relationship Diagramm

Balzert bewertet die Technik des Funktionsbaumes folgendermaßen:⁴⁰

Vorteil	Nachteil
- Bewährtes Konzept zur systematischen Gliederung von Funktionen	- Berücksichtigen nur die funktionale Sicht
- Gibt erste Hinweise für eine mögliche Dialoggestaltung	

Tabelle 8: Vor- und Nachteile eines Funktionsbaumes

⁴⁰ Balzert, „Lehrbuch der Software-Technik – Software Entwicklung“, Seite 117

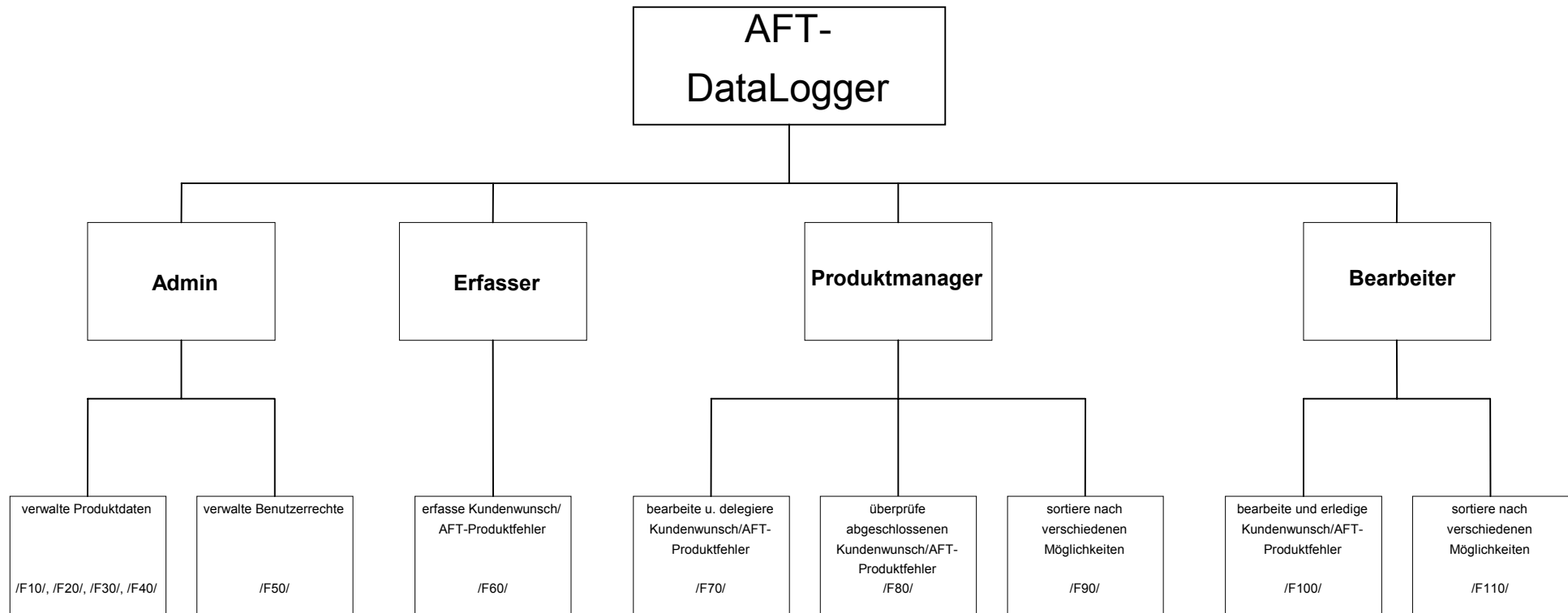


Abbildung 11: Funktionsbaum des AFT-DataLoggers

4.2.2 Datenflussdiagramm

Datenflussdiagramme wurden 1979 von DeMarco⁴¹ entwickelt. Bei einem Datenflussdiagramm werden die Wege von Daten bzw. Informationen zwischen den einzelnen Funktionen beschrieben. Bei der Entwicklung eines Datenflussdiagramms ist es wichtig sich vorzustellen, dass das zu entwickelnde System bereits läuft. Es ist zu vernachlässigen, wie das System initialisiert bzw. terminiert wird. Das Datenflussdiagramm wurde bereits 1966 unter der Bezeichnung Datenflusspläne in DIN 66001 genormt. In dieser Norm stehen 19 verschiedene Symbole zur Darstellung zur Verfügung. In der Praxis werden selten diese 19 Symbole verwendet. Meist wird die Notation von DeMarco verwendet, in der es vier Symbole gibt.

1. Datenfluss, dargestellt durch einen Pfeil mit Namen.
2. Funktion, dargestellt durch einen Kreis
3. Datenspeicher, dargestellt durch zwei Parallele Linien, zwischen denen der Speichername steht
4. Schnittstelle zur Umwelt, dargestellt durch ein Rechteck, das den Schnittstellennamen enthält

Beim erstellen von Datenfluss-Diagrammen definiert Balzert⁴² [2] die nachstehenden Regeln:

Syntaktische Regeln:

1. *Ein Datenfluss-Diagramm enthält mindestens eine Schnittstelle.*
2. *Jede Schnittstelle ist im allgemeinen nur einmal vorhanden. Wird durch diese Regel das Diagramm unübersichtlich, dann kann eine Schnittstelle auch mehrfach gezeichnet werden.*
3. *Zwischen Schnittstellen gibt es keine Datenflüsse.*
4. *Jeder Datenfluss hat einen Namen. Ausnahme: Datenflüsse, die zu Speichern führen oder dort beginnen und keinen Namen haben, transportieren die gesamten gespeicherten Daten.*
5. *Zwischen Speichern dürfen keine direkten Datenflüsse bestehen.*
6. *Zwischen Schnittstellen und Speichern dürfen keine direkten Datenflüsse gezeichnet sein.*

⁴¹ Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 138

⁴² Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 119f.

Semantische Regeln:

1. *Das DFD beschreibt den Datenfluss, nicht den Kontrollfluss. Daher enthält es weder Entscheidungen noch Schleifen. Es wird keine Aussage über die Initiierung und Terminierung von Funktionen bzw. Prozessen gemacht.*
 2. *Steht eine Schnittstelle für eine Vielzahl von beliebig vielen Instanzen, dann wird sie als eine Schnittstelle dargestellt.*
 3. *Wird das System durch eine eng begrenzte Anzahl von gleichartigen Schnittstellen begrenzt, die sich aber durch unterschiedliche Datenflüsse auszeichnen, dann ist eine getrennte Darstellung sinnvoll.*
 4. *Eine Schnittstelle ist so zu wählen, dass sie die ursprüngliche Quelle oder Senke einer Information angibt.*
 5. *Bei der Wahl einer Schnittstelle wird von der konkreten Eingabe einer Information in das System über eine Tastatur oder einer Ausgabe auf einem Drucker vollständig abstrahiert. Daher tauchen Tastatur und Drucker im allgemeinen nicht als Schnittstellen auf.*
 6. *Ein Datenflussname besteht aus einem Substantiv oder einem Adjektiv und einem Substantiv. Datenflussnamen enthalten niemals Verben.*
 7. *Die Datenflussnamen sind so zu wählen, dass sie nicht nur die Daten, die fließen, beschreiben, sondern etwas darüber aussagen, was über die Daten bekannt ist.*
 8. *Seichte Namen wie Daten, Informationen vermeiden.*
 9. *Ein Funktions- bzw. Prozessname besteht aus einem einzigen starken Aktions-Verb gefolgt von einem einzigen konkreten Objekt oder einem konkreten Substantiv gefolgt von einem starken Aktions-Verb*
 10. *Funktionsnamen repräsentieren Aktionen.*
 11. *Seichte Namen wie verarbeite, bediene vermeiden.*
-

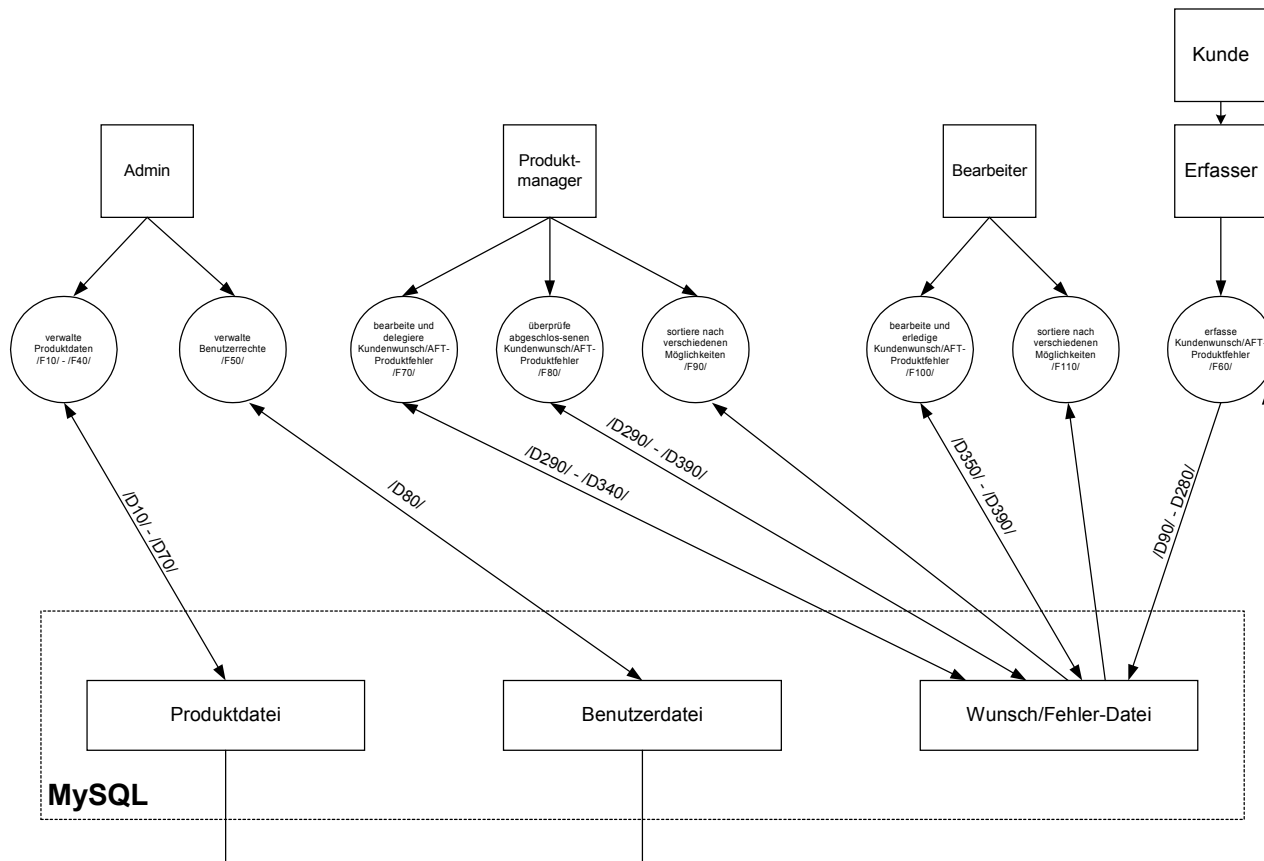


Abbildung 12: Datenfluß-Diagramm des AFT-DataLoggers

4.2.3 ER-Diagramm

Das Entity-Relationship-Diagramm (ER-Diagramm) wurde von P. Chen⁴³ 1976 zur Datenmodellierung entwickelt. Ziel eines ER-Diagramms ist es, die gespeicherten Daten und ihre Beziehungen untereinander zu beschreiben. Es entsteht ein konzeptionelles Modell, das gegen Veränderungen der Funktionalität weitgehend stabil ist.

Anwendungen erfordern die sichere Speicherung großer Datenmengen. Relationale Datenbanken sind zur Zeit in Sachen Datenbanksysteme führend. ER-Diagramme haben daher den großen Vorteil, dass das dort definierte Datenmodell direkt in einen relationalen Datenbankentwurf umgesetzt werden kann.

Ausgangspunkt eines jeden ER-Diagramms ist die Entität. *„Eine Entität ist ein individuelles und identifizierbares Exemplar von Dingen, Personen oder Begriffen der realen oder der Vorstellungswelt und wird durch Eigenschaften beschrieben.“*

⁴⁴Grafisch wird eine Entität durch ein Rechteck dargestellt, das den Namen der Entität enthält.

Beschrieben werden Entities durch ihre Attribute. Ein Attribut ist eine Eigenschaft, die die Entität genauer beschreibt. Schlüsselattribute sind Attribute, die die zugehörige Entität eindeutig identifizieren. Attribute werden in ER-Diagrammen meist als Ovale dargestellt und durch ungerichtete Kanten mit der jeweiligen Entitätsmenge verbunden. Manchmal werden die Attribute auch in das Rechteck der zugehörigen Entität geschrieben. Schlüsselattribute werden im ER-Diagramm unterstrichen dargestellt oder mit Schlüsselsymbolen gekennzeichnet.

Zwischen einzelnen Entitäten gibt es Beziehungen, auch „Relationships“ genannt. Beziehungen beschreiben einen Zusammenhang zwischen Entities, der im Allgemeinen durch ein Verb ausgedrückt wird. Grafisch werden Beziehungen als Raute dargestellt. In dieser Raute steht der Name der Beziehung. Die Raute verbindet durch Linien die über Beziehung verknüpften Entitäten. Die Raute kann auch weggelassen werden und der Name der Beziehung direkt an die Linie geschrieben werden.

⁴³ Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 138

⁴⁴ Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 138

Beziehung	Beschreibung
1-1-Beziehung	Zu jedem Objekt des Entity-Typs 1 gehört genau ein Objekt des Entity-Typs 2
1-n-Beziehung	Zu einem Objekt des Entity-Typs 1 gehören mehrere Objekte des Entity-Typs 2, zu jedem Objekt des Entity-Typs 2 lässt sich genau ein Objekt des Entity-Typs 1 finden.
m-n-Beziehung	Zu n Objekten des Entity-Typs 1 gehören m Objekte des Entity-Typs 2

Tabelle 9: Beziehungen zwischen Entitäten

Das ER-Diagramm wird mit der Software ERwin von Platinum erstellt. ERwin stellt die ER-Diagramme mit den folgenden Symbolen dar.

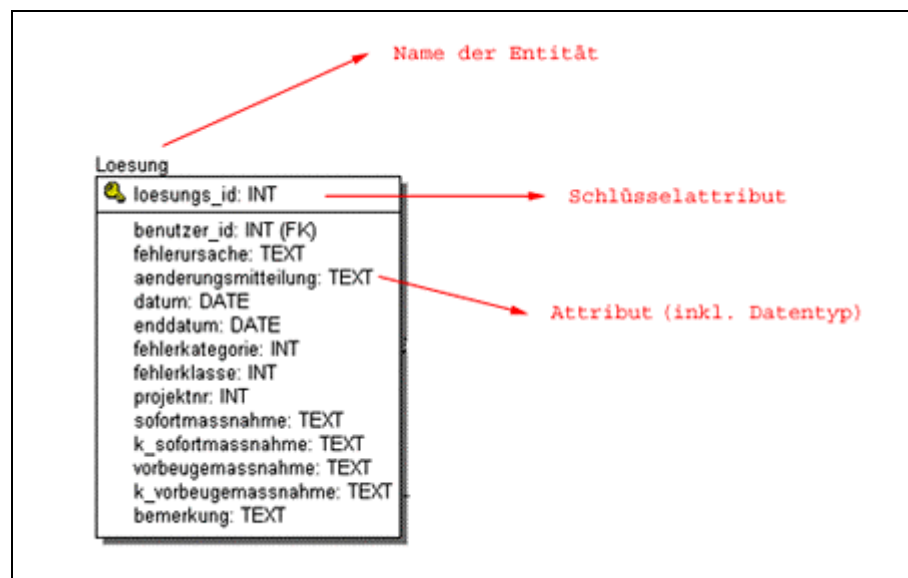


Tabelle 10: Grafische Darstellung in ERwin

Beziehungen in ERwin	Grafische Umsetzung in ERwin
1-1-Beziehung	-----
1-n-Beziehung	-----<{
m-n-Beziehung	{-----<{

Tabelle 11: Beziehungen zwischen Entitäten in ERwin

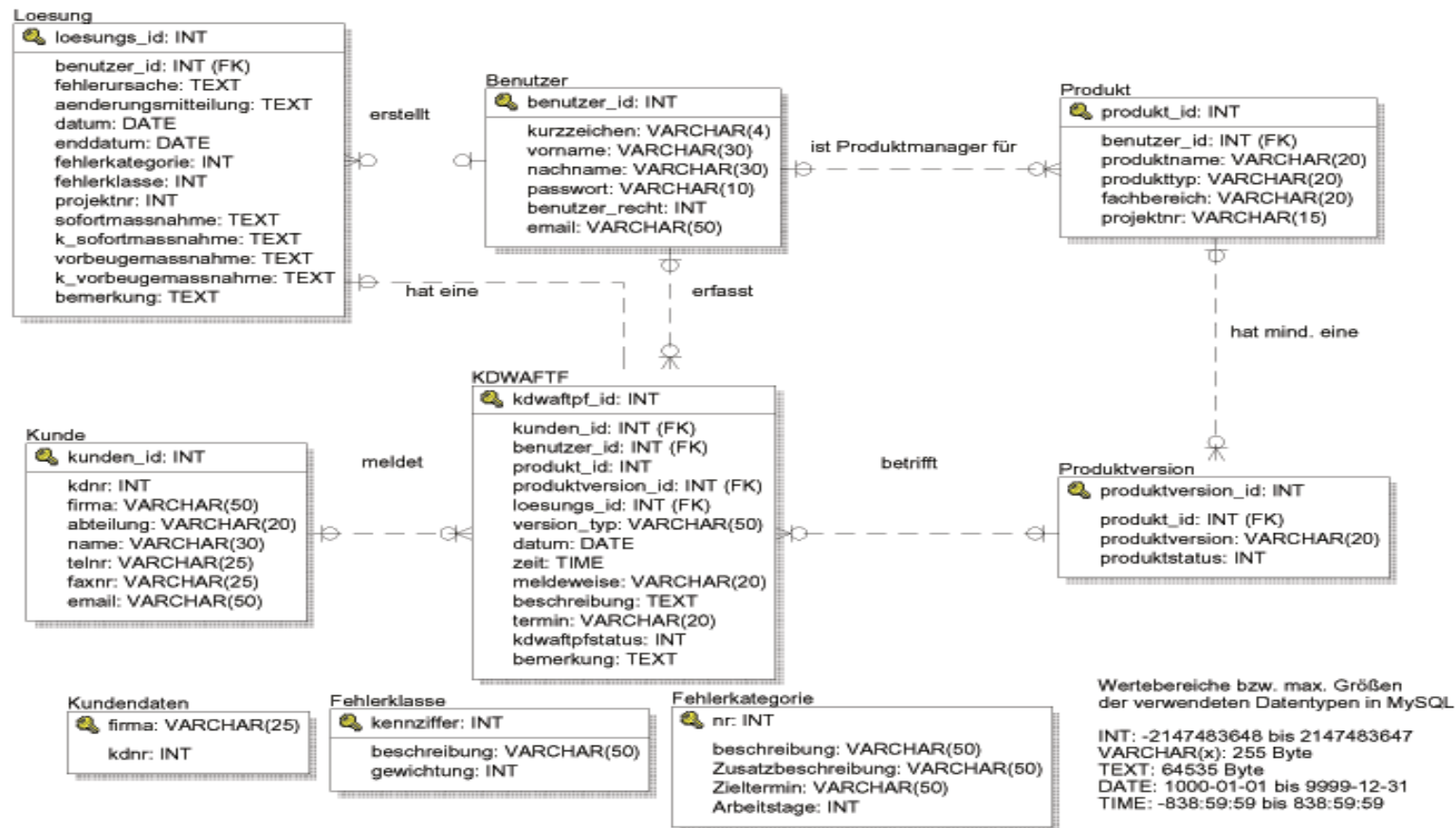


Abbildung 13: Das ER-Diagramm des AFT-DataLogger

4.3 Die Entwurfsphase – Phase 3

In der Entwurfsphase werden aus den gegebenen Anforderungen an ein Software-Produkt eine software-technische Lösung im Sinne einer Softwarearchitektur entwickelt. Das Entwerfen innerhalb dieser Entwurfsphase wird auch als „Programmieren im Großen“ bezeichnet. Ausgangspunkt ist das Pflichtenheft, in dem die Anforderungen definiert wurden. Sind zudem noch die Randbedingungen und Umgebungsbedingungen geklärt, festgelegt und Grundsatzentscheidungen bezüglich der Datenhaltung, Benutzungsoberfläche, Hilfesystem und Expertensysteme getroffen worden, kann mit der eigentlichen Entwurfsaktivität begonnen werden.

Die zu entwickelnde Softwarearchitektur wird dabei von verschiedenen Faktoren beeinflusst. Die Faktoren werden folgendermaßen unterscheiden.

- *Einsatzbedingungen, die sich aus den Produktanforderungen ergeben.*

Die Entscheidungen bezüglich der Einsatzbedingungen haben wesentlichen Einfluss auf die Softwarearchitektur und dem anzuwendenden Entwurfskonzept. Entscheidungen werden anhand der geplanten Einsatzbereiche getroffen, z.B. soll das Produkt sequentiell oder nicht sequentiell entworfen werden?

- *Umgebungs- und Randbedingungen, die sich durch die verwendete Zielplattform bzw. die verwendeten Zielplattformen ergeben.*

Anhand der Einsatzbedingungen ergibt sich meist, auf welchen Zielplattformen das Produkt später lauffähig sein muss. Auf dieser Zielplattform muss das Produkt dann in die bestehende Hardware-, Systemsoftware- und Anwendungssoftware eingebettet werden.

- *Randbedingungen, die sich aus nichtfunktionalen Produktanforderungen und Qualitätsanforderungen ergeben.*

Nichtfunktionale Produktanforderungen haben ebenfalls entscheidenden Einfluss auf die Softwarearchitektur, so muss z.B. bei einem international eingesetzten Produkt die Software-Architektur Anpassungen an landesspezifischen Funktionalitäten erlauben (Landeswährung, Sprache usw.).

Beeinflusst wird durch diese Faktoren nicht nur die Software-Architektur, sondern auch die weiter oben beschriebenen Grundsatzentscheidungen. Die folgende Abbildung gibt eine grafische Übersicht der Einflussfaktoren.

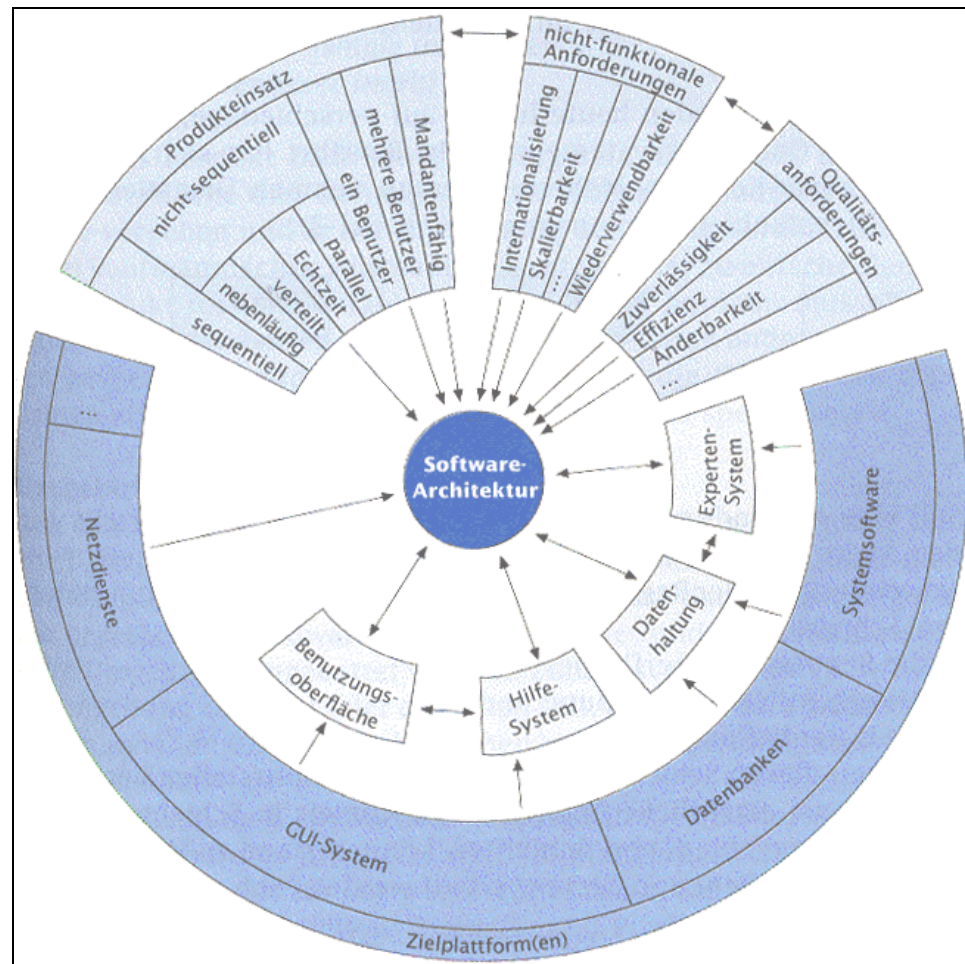


Abbildung 14: Einflussfaktoren auf die Software-Architektur⁴⁵

4.4 Die Implementierungsphase – Phase 4

In der Implementierungsphase werden aus den vorgegebenen Spezifikationen Programme entwickelt. Als Ausgangspunkt dienen die Ergebnisse der Entwurfsphase. Programmierung in der Implementierungsphase wird auch als „Programmieren im kleinen“ bezeichnet.

⁴⁵ Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 633

Folgende Tätigkeiten sind beim Implementieren zu tätigen:

- Konzeption von Datenstrukturen und Algorithmen
- Strukturierung des Programms
- Dokumentation der Problemlösung und der Implementierungsentscheidung
- Umsetzung der Konzepte in die Konstrukte der verwendeten Programmiersprache
- Angaben zur Zeit- und Speicherkomplexität des Programms
- Test oder Verifikation des entwickelten Programms

Am Ende dieser Tätigkeiten steht dann das Quellprogramm einschließlich integrierter Dokumentation, das Objektprogramm sowie die Tastplanung und Testprotokolle.

4.4.1 Prinzipien der Implementierung

Folgende Prinzipien sollten bei der Implementierung eingehalten werden:

- Prinzip der Verbalisierung
- Prinzip der problemadäquaten Datentypen
- Prinzip der Verfeinerung
- Prinzip der integrierten Dokumentation

4.4.2 Ziele der Implementierung

„Ziel der Implementierung ist es, die in der Softwarearchitektur spezifizierten Systemkomponenten durch Programme zu realisieren. Zur Implementierung der Systemkomponente werden in der Implementierungsphase

- *die Basiskonzepte*
 - *Kontrollstrukturen und*
 - *Entscheidungstabellen,*
- *Die Prinzipien*
 - *Verbalisierung,*
 - *Problemadäquate Datentypen,*
 - *Verfeinerung,*

- *Integrierte Dokumentation und*
- *Die Methoden*
 - *Schrittweise Verfeinerung*
 - *Selbstkontrolliertes Programmieren*

eingesetzt.

*Denkfallen und Programmierfallen können durch Beachtung, Anpassung und Weiterentwicklung eines Regelkataloges vermieden werden. Die Einhaltung von Programmierrichtlinien und Empfehlungen sorgt für einen einheitlichen Programmierstil.*⁴⁶

4.5 Die Abnahme- und Einführungsphase – Phase 5

In der Abnahme- und Einführungsphase wird das fertiggestellte Software-Produkt schließlich in Betrieb genommen.

4.5.1 Die Abnahmephase:

In der Abnahmephase kommt es zu verschiedenen Tätigkeiten:

- Das Gesamtprodukt wird einschließlich sämtlicher Dokumentationen an den Auftraggeber bzw. einer freigabeberechtigten Instanz innerhalb der Firma übergeben.
- Verbunden mit der Übergabe ist auch ein Abnahmetest. Im Rahmen dieser Abnahme werden verschiedene Tests durchgeführt, die normalerweise der Abnehmer ausgearbeitet hat und an denen er testet, ob das Produkt seinen Anforderungen entspricht.

Als Ergebnis der Abnahmephase steht das Abnahmeprotokoll. In diesem Protokoll werden sämtlichen relevanten Eingabedaten, durchgeführten Tests und erhaltene Ergebnisse festgehalten.

⁴⁶ Balzert, „Lehrbuch der Software-Technik – Software-Entwicklung“, S. 958

4.5.2 Die Einführungsphase:

Wenn es sich bei dem entwickelten Software-Produkt um ein auftragsbezogenes Produkt handelt, erfolgt nach der erfolgreichen Abnahme die Einführung des Produkts beim Auftraggeber. In dieser Einführungsphase werden folgende Tätigkeiten durchgeführt:

- Installation des Produkts (Einrichtung des Produkts in dessen Zielumgebung zum Zwecke des Betriebes)
- Schulung der Benutzer und des Personals
- Inbetriebnahme des Produkts (Übergang zwischen Installation und Betrieb)

Sämtliche Vorkommnisse bei der Einführungsphase werden in einem Einführungsprotokoll festgehalten. Umfangreiche Softwareeinführungen sind mit Innovationseinführungen gleichzusetzen.

Ist das Produkt erfolgreich eingeführt, wird es endgültig freigegeben und damit die Produktentwicklung offiziell abgeschlossen. Das Produkt wird dann in das Wartungsarchiv übernommen.

Als Endergebnis der Abnahme- und Einführungsphase stehen somit folgende Dinge:

- Gesamtprodukt einschließlich Gesamtdokumentation,
- Abnahmeprotokoll,
- Einführungsprotokoll

4.6 Die Wartungs- & Pflegephase – Phase 6

Diese Phase umfasst zwei Tätigkeiten, einmal die Wartung eines Produktes und zum anderen die Pflege eines Produktes. Wo liegen dort genau die Unterschiede?.

Balzert definiert diese zwei Begriffe folgendermaßen:

„Wartung beschäftigt sich mit der Lokalisierung und Behebung von Fehlerursachen bei in Betrieb befindlichen Software-Produkten, wenn die Fehlerwirkung bekannt ist“⁴⁷

„Pflege beschäftigt sich mit der Lokalisierung und Durchführung von Änderungen und Erweiterungen von in Betrieb befindlichen Software-Produkten, wenn die Art der gewünschten Änderungen/Erweiterungen festliegt“⁴⁸

Nachdem das Software-Produkt die Abnahme- und Einführungsphase durchlaufen hat, beginnt die Wartung und Pflege. Sobald das Produkt in Betrieb genommen wurde, treten im täglichen Betrieb Fehler auf, ändern sich die Umweltbedingungen (neue Hardware usw.) und es entstehen neue Wünsche und Anforderungen an das Produkt. Wird bei einer eingeführten Software keine Fehlerverbesserung durchgeführt, und Anpassungen sowohl an die Umwelt als auch an neue Anforderungen nicht vorgenommen, veraltet das Produkt sehr schnell und kann für den vorgesehenen Zweck nicht mehr verwendet werden.

In der Wartungs- und Pflegephase sind verschiedene Aktivitäten durchzuführen, die in vier Kategorien eingeteilt werden. Dies ersten beiden Kategorien umfassen dabei korrektive Tätigkeiten, während die letzten beiden Kategorien progressive Tätigkeiten umfassen.

1. Stabilisierung/Korrektur

Hierunter fallen Tätigkeiten, die sich mit der Fehlerbehebung befassen. Dies können Fehler sein, die bereits bei der Entwicklung oder während der Wartung entstanden sind.

2. Optimierung/Leistungsverbesserung

Die Optimierung bleibt meistens der Wartung vorbehalten, da Software-Produkte selten vor der ersten Freigabe optimiert sind. Sobald ein Produkt funktionsfähig ist, wird es meist direkt freigegeben. Tätigkeiten zur Optimierung sind all diejenigen, die dazu beitragen das

⁴⁷ Balzert, Lehrbuch der Software-Technik – Software-Entwicklung, S. 969

⁴⁸ Balzert, Lehrbuch der Software-Technik – Software-Entwicklung, S. 969

Produkt leistungsfähiger zu machen, wie z.B. Tuning, Monitoring und Reduzierung des Speicherbedarfes.

3. *Anpassung/Änderung*

Anpassungen und Änderungen entstehen durch geänderte Anforderungen der Umwelt, dies können u.a. Änderungen in der technischen Umgebung, in der Benutzeroberfläche oder in einzelnen Funktionen sein.

4. *Erweiterung*

Erweiterungen sind funktionale Ergänzungen eines Produktes. Dies können völlig neue Funktionen sein, die sich aus den Erfordernissen des Betriebs ergeben, oder welche die bei der Erstimplementierung noch nicht eingebaut wurden.

5 Software-Qualitätssicherung

Software-Produkte begegnen uns in vielen Bereichen unseres Lebens. In vielen Bereichen sind wir dabei besonders auf das fehlerfreie Funktionieren von Software-Systemen angewiesen, als Beispiel seien hier Flugzeuge und medizinische Geräte genannt, bei denen Softwarefehler oft Menschenleben kosten. Die Software-Entwicklung hat daher ein oberstes Gebot: Sie muss fehlerfrei sein. Bei Programmen, dessen Sourcecode sich auf einige wenige Zeilen beschränken, ist dies noch ohne weiteres machbar, aber bei komplexen Programmen mit mehreren Tausend Programmzeilen ist dies ein schwieriges Unterfangen. Zu diesem Zweck gibt es die Software-Qualitätssicherung, die sich darum bemüht, Fehlerfreien Code zu programmieren. In diesem Kapitel werden die Grundlagen der Software-Qualitätssicherung erläutert, um einen Eindruck zu vermitteln, wie komplex dieser Bereich der Software-Entwicklung ist.

Die Unterkapitel im einzelnen

- Definition Qualität und Software-Qualität
- Qualitätsmodelle
- Qualitätszielbestimmung

5.1 *Definition: Qualität und Software-Qualität*

Fehlerfreiheit ist ein Qualitätsaspekt eines Software-Produkts und der Begriff der Qualität ist zunächst zu definieren. Es gibt fünf verschiedene betriebliche Sichten auf ein Produkt, so dass 5 Ansätze unterschieden werden können:

- der transzendente Ansatz
- der produktbezogene Ansatz (Entwicklung)
- der benutzerbezogene Ansatz (Marketing/Vertrieb)
- der prozessbezogene Ansatz (Fertigung)
- der Kosten/Nutzen-bezogene Ansatz (Finanzen)

Nach DIN 55350, Teil 11 ist Qualität entsprechend dem produkt- und prozessbezogenem Ansatz folgendermaßen definiert:

„Qualität ist die Gesamtheit von Eigenschaften und Merkmalen eines Produktes oder einer Tätigkeit, die sich auf deren Eignung zur Erfüllung gegebener Erfordernisse bezieht.“

Nach DIN ISO 9126 ist Software-Qualität folgendermaßen definiert:

„Software-Qualität ist die Gesamtheit der Merkmale und Merkmalswerte eines Software-Produkts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen.“

5.2 Qualitätsmodelle

Diese eher allgemeine Definition von Software-Qualität ist für praktische Anwendungen oft nicht ausreichend, so dass die Software-Qualität durch ein Qualitätsmodell ergänzt wird. Ein Qualitätsmodell präzisiert den Qualitätsbegriff durch Ableiten von Unterbegriffen. Dabei wird die Software-Qualität durch Qualitätsmerkmale beschrieben, welche in weiteren Schritten in Teilmerkmale verfeinert werden. Im Englischen Sprachraum wird anstelle des Begriffs Merkmal der Begriff factor verwendet. Ein so aufgebautes Modell wird als FCM-Modell (factor-criteria-metrics-model) bezeichnet. Folgende Abbildung zeigt die Software-Qualitätsmerkmale, die in ISO 9126 definiert sind.

Funktionalität

Vorhandensein von Funktionen mit festgelegten Eigenschaften. Diese Funktionen erfüllen die definierten Anforderungen.

- Richtigkeit
- Angemessenheit
- Interoperabilität
- Ordnungsmäßigkeit
- Sicherheit

Zuverlässigkeit

Fähigkeit der Software, ihr Leistungsniveau unter festgelegten Bedingungen über einen festgelegten Zeitraum zu bewahren.

- Reife
- Fehlertoleranz
- Wiederherstellbarkeit

Benutzbarkeit

Aufwand, der zur Benutzung erforderlich ist, und individuelle Beurteilung der Benutzung durch eine festgelegte oder vorausgesetzte Benutzergruppe.

- Verständlichkeit
- Erlernbarkeit
- Bedienbarkeit

Effizienz

Verhältnis zwischen dem Leistungsniveau der Software und dem Umfang der eingesetzten Betriebsmittel unter festgelegten Bedingungen.

- Zeitverhalten
- Verbrauchsverhalten

Änderbarkeit

Aufwand, der zur Durchführung vorgegebener Änderungen notwendig ist. Änderungen können Korrekturen, Verbesserungen oder Anpassungen an Änderungen der Umgebung, der Anforderungen und der funktionalen Spezifikationen einschließen.

- Analysierbarkeit
- Modifizierbarkeit
- Stabilität
- Prüfbarkeit

Überprüfbarkeit

Eignung der Software, von einer Umgebung in eine andere übertragen zu werden. Umgebung kann organisatorische Umgebung, Hardware- oder Software-Umgebung einschließen.

- Anpassbarkeit
- Installierbarkeit
- Konformität
- Austauschbarkeit

Tabelle 12: Software-Qualitätsmerkmale nach DIN ISO 9126

5.3 Qualitätszielbestimmung

Die Qualitätszielbestimmung muss vor Entwicklungsbeginn durchgeführt werden. Das Ergebnis einer solchen Bestimmung wird im Pflichtenheft festgehalten. Bei der Verwendung von Qualitätsmodellen die sich auf Indikatoren stützt, müssen Qualitätsstufen definiert werden. Sind Qualitätsstufen definiert, muss definiert sein, welche Qualitätsstufen erreicht werden sollen

5.3.1 Qualitätsstufe

DIN ISO 9126 definiert die Qualitätsstufe folgendermaßen:

„Eine Qualitätsstufe ist ein Wertebereich auf einer Skala, mit deren Hilfe Software entsprechend den festgelegten Qualitätsanforderungen klassifiziert werden kann“

5.3.2 Qualitätsanforderungen

Aus der Qualitätszielbestimmung entstehen Qualitätsziele, aus denen sich dann Qualitätsanforderungen ergeben. Die Qualitätsanforderungen legen fest, welche Qualitätsmerkmale im konkreten Fall als relevant erachtet werden und in welcher Qualitätsstufe sie erreicht werden sollen.

5.3.3 Qualitätszielbestimmung des AFT-DataLogger

Bei der Erstellung des Pflichtenheftes wurde auch eine Qualitätszielbestimmung durchgeführt. Die folgende Tabelle ist ein Auszug aus dem Pflichtenheft des AFT-DataLogger.

Kursiv und fett markierter Bereich ist die für das entsprechende Teilmerkmal relevante Produktqualität

Qualitätsmerkmale	Merkmaldefinition	Produktqualität			
		Sehr gut	Gut	Normal	Nicht relevant
Funktionalität					
Angemessenheit	Eignung der Funktionen für spezifizierte Aufgaben, z.B. aufgabenorientierte Zusammensetzung von Funktionen aus Teilfunktionen	<i>Jede Funktion erfüllt nur die ihm zugedachten Aufgaben</i>		Funktionen können mehrere Verarbeitungsschritte in sich verbergen	
Richtigkeit	Liefern der richtigen oder vereinbarten Ergebnisse oder Wirkungen, z.B. die benötigte Genauigkeit von berechneten Werten	<i>Die Funktionen liefern exakt die geforderten Ergebnisse oder Wirkungen</i>		Die Funktionen liefern Ergebnisse, die Funktionalität des Programms nicht beeinträchtigen	
Interoperabilität	Fähigkeit, mit vorgegebenen Systemen zusammenzuwirken	Das Programm arbeitet auf allen im Betrieb verwendeten Systeme	<i>Das Programm arbeitet auf die im Pflichtenheft definierten Systemen</i>	Das Programm ist nur auf ein bestimmtes System einsetzbar	
Ordnungsmäßigkeit	Erfüllung von anwendungsspezifischen Normen, Vereinbarungen, gesetzlichen Bestimmungen und ähnlichen Vorschriften	Das Programm erfüllt alle Anforderungen	<i>Das Programm erfüllt die im Pflichtenheft angegebenen Anforderungen</i>		
Sicherheit	Fähigkeit, unberechtigten Zugriff, sowohl versehentlich als auch vorsätzlich, auf Programme und Daten zu verhindern	<i>Das Programm verbietet unberechtigten Zugriff durch Benutzer-authentifikation</i>			
Zuverlässigkeit					
Reife	Geringe Versagenshäufigkeit durch Fehlzustände	Jede noch so erdenkliche Fehlermöglichkeit wird im Programm durch eine Fehleroutine abgefangen, um ein Versagen des Programms zu verhindern	<i>Durch die definierten Testfälle werden die Spezifikationen getestet</i>		

Fehlertoleranz	Fähigkeit, ein spezifiziertes Leistungsniveau bei Software-Fehlern oder Nicht-Einhaltung ihrer spezifizierten Schnittstelle zu bewahren	Eingabefelder werden auf einen erwarteten Datentyp geprüft.		
Wiederherstellbarkeit	Fähigkeit, bei einem Versagen das Leistungsniveau wiederherzustellen und die direkt betroffenen Daten wiederzugewinnen. Zu berücksichtigen sind die dafür benötigte Zeit und der benötigte Aufwand	Jedes eingetippte Zeichen des Benutzers wird sofort in der Datenbank gesichert und die Datenbank wird bei jedem neuen Datensatz gesichert	Nach betätigen eines definierten Button wird der Datensatz in der Datenbank gespeichert und die Datenbank wird bei jedem neuen Datensatz gesichert	Nach betätigen eines definierten Button wird der Datensatz in der Datenbank gespeichert. Die Datenbank wird in einem vorher definierten Zeitintervall gesichert
Benutzbarkeit				
Verständlichkeit	Aufwand für den Benutzer, das Konzept und die Anwendung zu verstehen	Das Konzept und die Anwendung ist auf dem ersten Blick klar erkennbar	Das Konzept und die Anwendung ist nach sehr kurzer Zeit klar verständlich	Das Konzept und die Anwendung ist erst nach einer Schulung verständlich
Erlernbarkeit	Aufwand für den Benutzer, die Anwendung zu erlernen (z.B. Bedienung, Ein- Ausgabe)	Die Anwendung bedarf keinerlei Aufwand	Die Anwendung bedarf einem kurzen Lernaufwand	Die Anwendung bedarf einer Schulung
Bedienbarkeit	Aufwand für den Benutzer, die Anwendung zu bedienen	Die Bedienung bedarf keinem Aufwand	Die Bedienung bedarf einer kurzen Einarbeitungszeit	Die Bedienung bedarf einer Schulung
Effizient				
Zeitverhalten	Antwort- und Verarbeitungszeiten sowie Durchsatz bei der Funktionsausführung	Bei Abfragen der Datenbank entstehen keinerlei Verarbeitungszeiten. Die gröÙe der Datenbank spielt dabei keine Rolle	Bei Abfragen der Datenbank entstehen je nach Datenbankgröße kurze Verarbeitungszeiten	
Änderbarkeit				
Analysierbarkeit	Aufwand, um Mängel oder Ursachen von Versagen zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen	Jedes Modul und jede Funktion sind gut erläutert so dass sich der Aufwand der Analysierbarkeit in Grenzen hält		

Modifizierbarkeit	Aufwand zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder Anpassung an Umgebungsänderungen	<i>Durch die gute Dokumentation der Module und Funktionen ist der Aufwand minimal</i>	
Stabilität	Wahrscheinlichkeit des Auftretens unerwarteter Wirkungen von Änderungen	<i>Sind auszuschließen nach Wiederholung sämtlicher Testfälle</i>	Nur die geänderte Funktionen werden getestet
Prüfbarkeit	Aufwand, der zur Prüfung der geänderten Software notwendig ist	<i>Der Aufwand ist gering, da die einzelnen Module und Funktionen gut erläutert sind</i> <i>Es muss nur für die neuen Funktionen neue Testfälle generiert werden</i>	
Übertragbarkeit			
Anpassbarkeit	Möglichkeiten, die Software an verschiedene, festgelegte Umgebungen anzupassen, wenn nur Schritte unternommen oder Mittel eingesetzt werden, die für diesen Zweck für die betrachtete Software vorgesehen sind		<i>Das Programm wird nur wie im Pflichtenheft definiert, auf eine Umgebung angepasst</i>
Installierbarkeit	Aufwand, der zum installieren der Software in einer festgelegten Umgebung notwendig ist		<i>Das Programm bedarf keiner Installation auf der Anwenderseite</i>
Konformität	Grad, in dem die Software Normen oder Vereinbarungen zur Übertragbarkeit erfüllt		<i>Übertragbarkeit ist im Pflichtenheft nicht definiert</i>
Austauschbarkeit	Möglichkeit, diese Software anstelle einer spezifizierten anderen in der Umgebung jener Software zu verwenden, sowie der dafür notwendige Aufwand		<i>Diese Möglichkeit ist im Pflichtenheft nicht definiert</i>

Tabelle 13: Qualitätszielbestimmung des AFT-DataLogger

6 Anwendung: AFT- DataLogger

Mit Hilfe des AFT-DataLogger können eingehende Kundenwünsche oder AFT-Produktfehler erfasst werden. Durch das Einbinden des Programms in das Intranet von AFT ist eine Erfassung im gesamten Firmenbereich gewährleistet. So kann jeder Mitarbeiter von AFT, der Zugang zum Intranet hat, eingehende Datensätze erfassen. Sobald ein Datensatz erfasst wurde, erhält der zuständige Produktmanager eine E-Mail, in der ihm der Eingang des neuen Datensatzes mitgeteilt wird. Seine Aufgabe besteht nun darin, diesen neu eingegangenen Datensatz zuzuordnen, und einen Bearbeiter auszuwählen, der ihn bearbeitet. Dieser ausgewählte Bearbeiter erhält nach der Zuordnung ebenfalls eine E-Mail. Der Bearbeiter kann sich nun im System den eingegangenen Datensatz inklusive der erfassten Daten anschauen und schließlich bearbeiten. Sobald die Bearbeitung abgeschlossen ist, wird der Produktmanager per E-Mail über den Abschluss informiert.

In diesem Kapitel wird der AFT-DataLogger vorgestellt. Anfangs erfolgt zunächst eine kurze Erläuterung über die Benutzerrechtvergabe sowie die Einbindung des Programms in das bestehende Intranet-System. Anschließend folgt die Präsentation des Programms gegliedert nach den einzelnen Benutzerrechten.

Die Unterkapitel im einzelnen:

- Die Benutzerrechte im System
- Die Benutzerrechtvergabe
- Allgemeiner Seitenaufbau
- Startpunkt „YaWohl!“
- Login
- Hauptmenu
- Datenbankzugriff

6.1 Die Benutzerrechte im System

Im AFT-DataLogger gibt es vier Benutzerrechte.

- | | |
|--------------------|---|
| 1. Erfasser: | Standardmäßig ist jeder Mitarbeiter der Firma AFT Erfasser. Der Erfasser hat nur die Berechtigung, eingehende Kundenwünsche, bzw. „AFT-Produktfehler“ zu erfassen. |
| 2. Bearbeiter: | Ein Bearbeiter hat das Recht, eingegangene Datensätze zu bearbeiten. Der Bearbeiter wird vom Produktmanager bestimmt. Der Bearbeiter kann nur die ihm zugewiesenen Datensätze bearbeiten. |
| 3. Produktmanager: | Produktmanager haben das Recht, eingegangene Kundenwünsche/AFT-Produktfehler zuzuordnen. Dabei wird der eingegangene Datensatz u.a. einer Fehlerkategorie zugeordnet und einem Bearbeiter zugewiesen. |
| 4. Administrator: | Der Administrator kann Benutzer anlegen, löschen, Benutzerdaten ändern, Produkte anlegen und verwalten. |

Die Rechte sind in sich kombinierbar, d.h. ein Administrator kann gleichzeitig auch Produktmanager sein, oder ein Produktmanager kann zusätzlich auch Bearbeiter sein und gleichzeitig noch Administrator. Klar sein muss, dass alle User mindestens Erfasserrechte haben.

6.2 Die Benutzerrechtvergabe

Wird ein Benutzer in der Datenbank angelegt, erhält dieser automatisch Erfasserrechte. Der Benutzer kann weitere Rechte auf folgende Art und Weise erhalten:

- Der Benutzer bekommt das Recht „Produktmanager“ sobald er im System für ein Produkt als Produktmanager ausgewählt wurde. Die-

ses Recht kann ihm der Administrator zuteilen, aber auch wieder entziehen.

- Er erhält das Recht „Bearbeiter“, sobald er als Bearbeiter für einen Datensatz ausgewählt wurde. Wurde ihm einmal dieses Recht zugeteilt, kann es ihm nicht mehr entzogen werden, auch wenn für ihn keine Datensätze mehr zum Bearbeiten anstehen.
- Das „Administrator-Recht“ kann diesem nur von einem bereits existierenden Administrator zugeteilt und wieder entzogen werden.

6.3 Allgemeiner Seitenaufbau

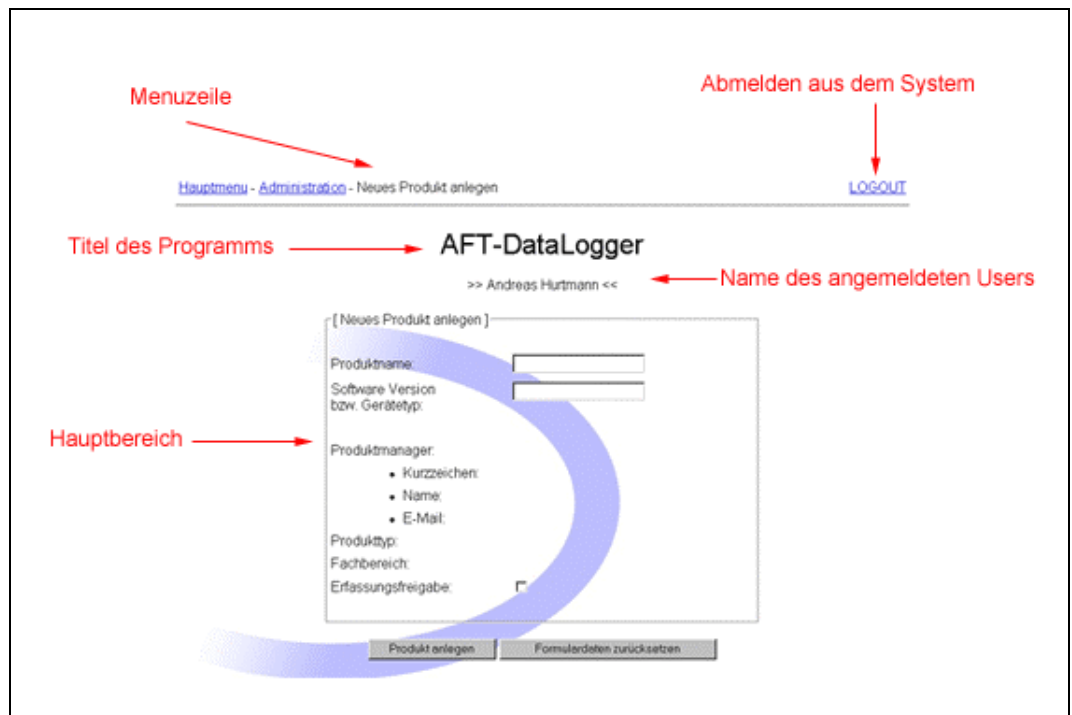


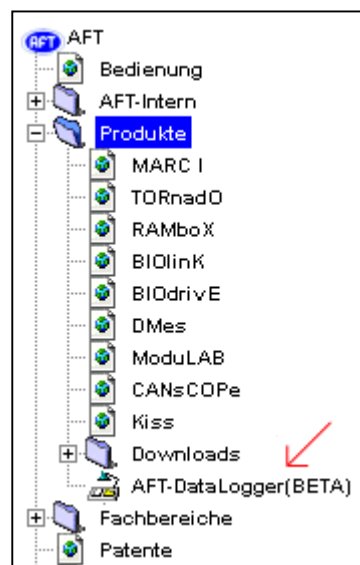
Abbildung 15: Allgemeiner Seitenaufbau des AFT-DataLogger

Jede Seite des AFT-DataLogger besteht aus den oben markierten Bereichen. Im oberen Bereich der Seite befindet sich zum einen die Navigation und zum anderen der Link zum Ausloggen aus dem System. Es ist auf jeder Seite und zu jeder Zeit möglich, sich aus dem System abzumelden. Unterhalb dieser Navigation befindet sich der Programmname sowie der Name des aktuell angemeldeten Benutzers. Im Hauptbereich befinden sich die entsprechenden Formulare des AFT-DataLogger.

6.4 Startpunkt „YaWohl!“

Der AFT-DataLogger ist ein webbasiertes-Tool zur Erfassung und Verfolgung von Kundenwünschen und AFT-Produktfehlern. Startpunkt ist „Ya-wohl!“, dem Intranetangebot von AFT!

Im Navigationsbaum von YaWohl! befindet sich der Link zum AFT-DataLogger unter dem Punkt „Produkte“. Mit einem klick auf **AFT-DataLogger** gelangt man zur Login-Seite.



6.5 Login

Um einen User im System identifizieren und authentifizieren zu können, muss sich der User mit einem Usernamen im System anmelden. Der Username entspricht dabei dem Mitarbeiterkurzzeichen der AFT. Die Eingabe eines Passwortes ist **nicht** erforderlich, wenn der User lediglich Erfasserrechte hat. Er gelangt nach Eingabe seines Usernamens sofort ins Erfassungsformular. User, die außer den Erfasserrechten weitere Rechte im System haben, werden nach Eingabe Ihres Usernamens nach einem Passwort gefragt. Ist das Passwort korrekt, gelangt der User ins Hauptmenu. Von dort aus hat er Zugang zu den für Ihn freigegebenen Funktionen. Die folgenden zwei Abbildungen zeigen die beiden Login-Seiten:

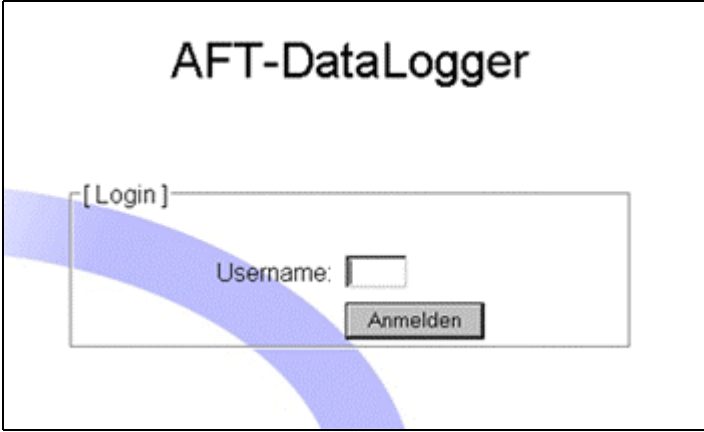


Abbildung 16: Login des AFT-DataLogger (1)

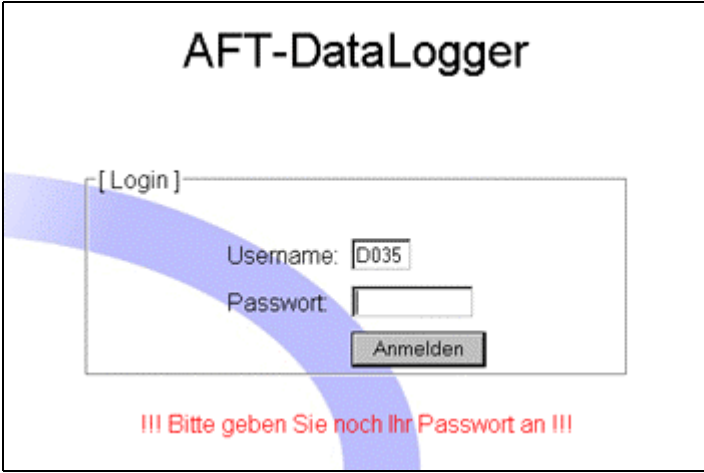


Abbildung 17: Login des AFT-DataLogger (2)

6.6 Hauptmenu

Ins Hauptmenu gelangen nur die User, die außer den Erfasserrechten weitere Rechte im System besitzen. Der User erkennt anhand der Links im Hauptmenu, welche Funktionen er ausführen darf. Für ihn nicht freigegebene Funktionen sind als reiner Text dargestellt.

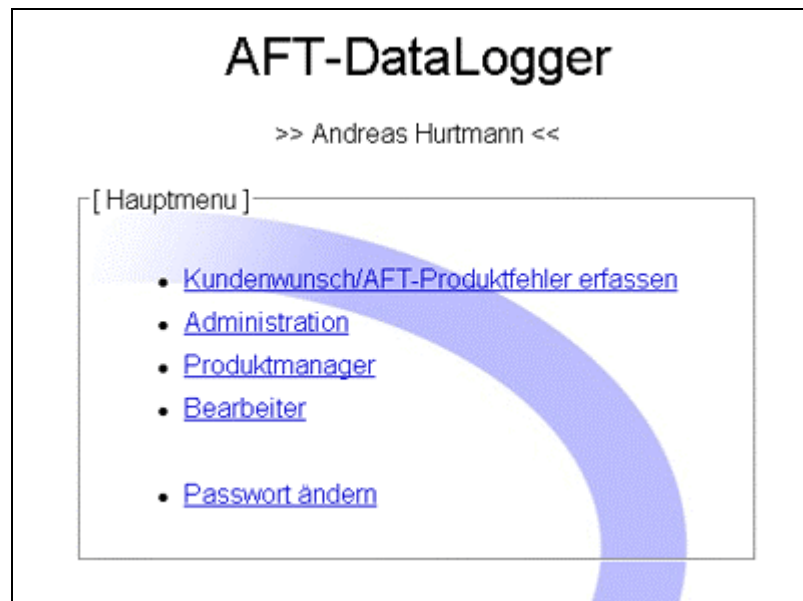


Abbildung 18: Das Hauptmenu des AFT-DataLogger für einen User der alle Berechtigungen besitzt

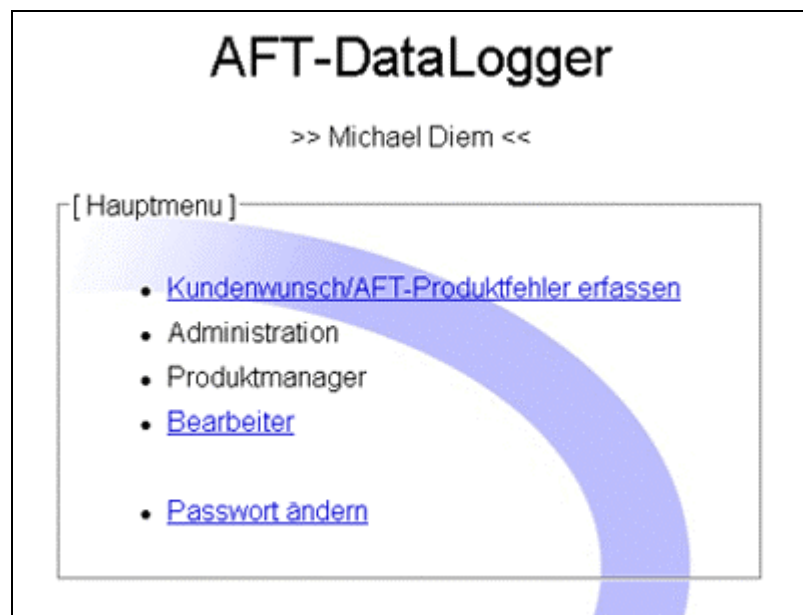



Abbildung 19: Hauptmenu des AFT-DataLogger für einen User mit 'Bearbeiter'-Rechten

6.6.1 Kundenwunsch/AFT-Produktfehler erfassen

In dieses Erfassungsformular, gelangt man entweder über das Hauptmenu, oder nach Eingabe des Usernamens beim Login, wenn man nur Erfassungsrechte besitzt. Hier werden die zu erfassenden Daten aufgenommen, die nötig sind, um den Kundenwunsch bzw. den AFT-Produktfehler zu bearbeiten. Das Erfassungsformular ist dabei in drei Punkte aufgliedert. Allgemeine Daten, Kundendaten und Produktdaten.

- *Erfassung der Allgemeinen Daten:*



Allgemeine Daten

Erfasser: DI

Datum: 2001-12-15

Zeit: 18:41

Meldeweise:

Abbildung 20: Erfassung der Allgemeinen Daten

Die folgende Tabelle definiert die Erfassten Daten:

Erfasser:	Automatische Übernahme des Login-Namen
Datum:	Aktuelles Tagesdatum im Format <i>yyyy-mm-dd</i>
Zeit:	Aktuelle Zeit beim Aufruf des Formulars im Format <i>hh:mm</i>
Meldeweise:	Auswahlbox: Zur Auswahl stehen Telefon, Telefax, Email, Post und Sonstiges. Hier wird ausgewählt, auf welche Weise die Daten gemeldet wurden. Nur wenn die Art unter den aufgeführten Möglichkeiten nicht vorhanden ist, ist „Sonstiges“ zu wählen

Tabelle 14: Erfassungsfelder der Allgemeinen Daten

- Erfassung der Kundendaten:

Tabelle 15: Erfassungsfelder der Kundendaten

Die folgende Tabelle definiert die erfassten Daten:

**Kunden-Nr.:	Ist die Kundennummer des Kunden bekannt, wird sie eingegeben. Durch drücken auf den Button „Zuordnen“ wird die zur Kundennummer zugehörige Firma in der darunter liegenden Auswahlbox selektiert.
**Firma auswählen <u>oder</u> eingeben:	Ist die Kundennummer nicht bekannt, kann hier aus der Auswahlliste die Firma direkt ausgewählt werden. Die zugehörige Kundennummer wird automatisch im darüber liegenden Feld angezeigt. Ist die Firma nicht in der Liste enthalten, wird in der darunter stehenden Eingabezeile der Firmenname manuell eingegeben.
**Ansprechpartner:	Name der Person, der diesen Kundenwunsch/AFT-Produktfehler gemeldet hat.
**Telefon:	Telefonnummer, unter der der Ansprechpartner zu erreichen ist.

WICHTIG: Eines der beiden Felder <i>Ansprechpartner</i> oder <i>Telefon</i> muss ausgefüllt sein. Ansonsten kann der Datensatz nicht gespeichert werden.	
Abteilung:	Die Abteilung, in der der Ansprechpartner arbeitet.
Telefax:	Die Telefaxnummer der Firma oder des Ansprechpartners.
E-Mail:	Die E-Mail Adresse, unter der man den Ansprechpartner erreichen kann. Die E-Mail Adresse muss folgendem Format entsprechen: hans.mustermann@musterfirma.de mustermann@musterfirma.de fritz.mustermann@musterfirma.com usw.
Bemerkung:	Hier wird alles eingetragen, was noch relevant ist, um eine reibungslose Bearbeitung zu unterstützen. Bsp: Verweis auf bereits gemeldete Fehler, Angabe einer Handy-Nummer usw.

****:** Pflichtfelder

Tabelle 16: Erfassungsfelder der Kundendaten

- *Erfassung der Produktdaten:*

Abbildung 22: Erfassungsdaten der Produktdaten

Die folgende Tabelle definiert die Erfassten Daten:

Produktname:	Auswahlbox zur Auswahl des betreffenden Produktes.
Software Version / Gerätetyp	Nach Auswahl des Produktnamens stehen je nach ausgewähltem Produkttyp die möglichen Software-Versionen bzw. Gerätetypen des ausgewählten Produktes zur Verfügung.
Revision, Service Packs/Seriennummer:	Je nach ausgewähltem Produkttyp wird in diesem Eingabefeld die Revision oder das Service Pack bzw. die Seriennummer eingetragen.
Produkttyp, Produktmanager, Fachbereich, E-Mail:	Bereits nach Auswahl des Produktnamens stehen hier die Informationen über den Produkttyp (Software/Hardware), dem zuständigen Produktmanager, dessen Fachbereich und die E-Mail-Adresse des Produktmanagers.
Beschreibung:	Hier wird die Beschreibung des Kundenwunsches/AFT-Produktfehlers eingetragen. Sonstige Zusatzbemerkungen werden im Feld „Bemerkungen“ unter Kundendaten eingetragen.
Termin:	In dieser Eingabezeile werden evtl. Terminwünsche eingetragen.

Tabelle 17: Erfassungsdaten der Produktdaten

- *Erfasste Daten speichern:*

Mit dem Drücken auf den Button „Daten in die Datenbank speichern“ erhält der User zur Kontrolle eine Übersicht über die eingegebenen Daten. Die folgende Abbildung zeigt diese Übersicht.

[Folgende Daten wurden erfasst]

Bitte überprüfen Sie die erfaßten Daten, bevor Sie in der Datenbank gespeichert werden

Allgemeine Daten:	Kundendaten:	Produktdaten:
• Erfasser: DI	• Firma: AISIN, Japan	• Produktname: MARC
• Meldeweise: Telefon	• Kunden-Nr.: 234	• Software Version: III
• Datum: 2001-12-15	• Ansprechpartner: Herr Mustermann	• Revision bzw. Service Pack: SP 4
• Zeit: 18:51	• Abteilung:	• Produkttyp: Software
	• Telefon:	• Produktmanager: Andreas Hurtmann
	• Telefax:	• Fachbereich: ECT
	• EMail:	• EMail: a.hurtmann@aft-werdohl.de
	• Bemerkung:	• Termin:
		• Beschreibung:

Abbildung 23: Erfassungsübersicht

Sind die Daten überprüft und für korrekt befunden, können Sie mit drücken auf den Button „Daten speichern“ in die Datenbank gespeichert werden. Es folgt nun noch der Hinweis, das die Daten in die Datenbank geschrieben wurden, und das der zuständige Produktmanager per E-Mail informiert wurde. Abbildung 24 zeigt diesen Hinweis. Sollte Korrekturbedarf bestehen, so besteht die Möglichkeit über den Button „Zurück zum Erfassungsformular“ die erfassten Daten nochmals zu editieren.

AFT-DataLogger

>> Andreas Hurtmann <<

[Kundenwunsch/AFT-Produktfehler erfassen]

Die Daten wurden erfolgreich in der Datenbank gespeichert, und der zuständige Produktmanager wurde per E-Mail informiert

Abbildung 24: Hinweis auf Speicherung der Daten

6.6.2 Administration

Dem Administrator stehen die in der folgenden Abbildung dargestellten Funktionen zur Verfügung. Diese lassen sich grob in die beiden Bereiche „Benutzerverwaltung“ und „Produktverwaltung“ untergliedern.

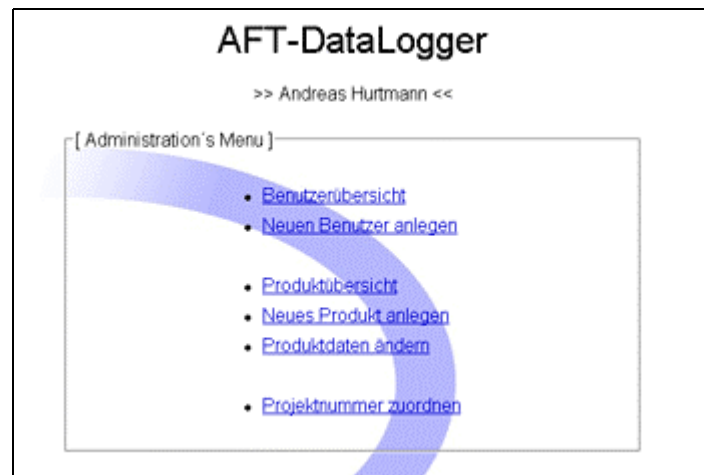


Abbildung 25: Hauptmenu des Administrator

- *Benutzerübersicht:*

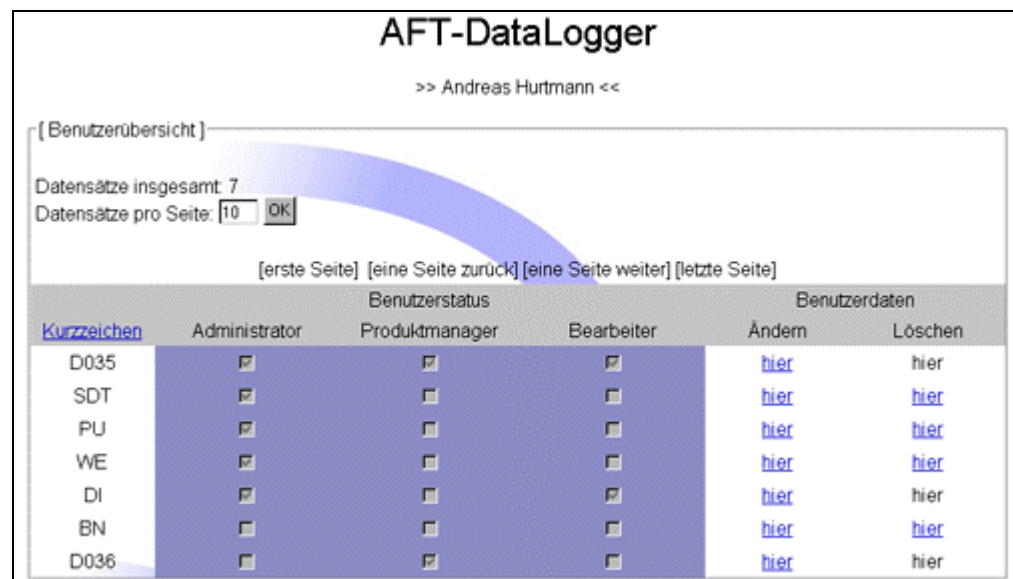


Abbildung 26: Benutzerübersicht

In diesem Unterpunkt erhält der Administrator eine Übersicht über alle im System eingetragenen Benutzer. Neben dem jeweiligem Kurzzeichen stehen die Rechte die dieser User im System hat. Des weiteren können hier Benutzer aus dem System gelöscht und bestehende Benutzerdaten geändert werden.

- *Benutzerdaten ändern:*

The screenshot shows the 'AFT-DataLogger' interface. At the top, it says '>> Andreas Hurtmann <<'. Below this is a section titled '[Benutzerdaten ändern]'. Inside this section, there are several fields: 'Kurzzeichen:' with the value 'D035', 'Vorname:' with 'Andreas', 'Nachname:' with 'Hurtmann', 'E-Mail:' with 'a.hurtmann@aft-werdohl.de', and 'Administrator:' with a checked checkbox. At the bottom of the form, there are two buttons: 'Änderungen speichern' and 'Formulardaten zurücksetzen'.

Abbildung 27: Benutzerdaten ändern

Hier hat der Administrator die Möglichkeit, dem ausgewählten Benutzer Administrator-Rechte zu verleihen oder zu entziehen. Neben dieser Möglichkeit können hier der Vorname, Nachname und die E-Mail-Adresse geändert werden.

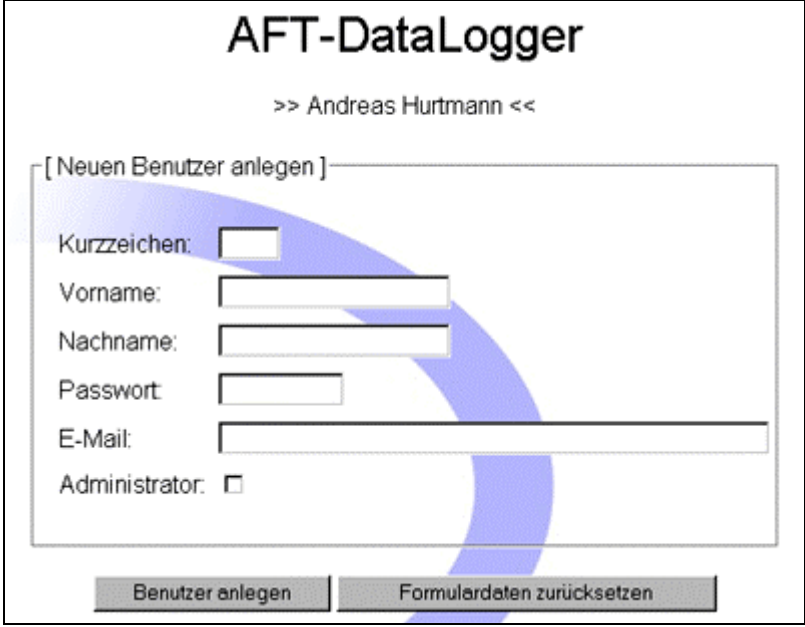
- *Benutzerdaten löschen:*

The screenshot shows the 'AFT-DataLogger' interface. At the top, it says '>> Andreas Hurtmann <<'. Below this is a section titled '[Benutzer löschen]'. Inside this section, there are several fields: 'Kurzzeichen:' with the value 'SDT', 'Vorname:' with 'Alfred', 'Nachname:' with 'Schmidt', 'E-Mail:' with 'a.schmidt@aft-werdohl.de', and 'Benutzerrechte:' with a checked checkbox and the label 'Administrator'. At the bottom of the form, there is a button labeled 'Benutzer löschen'.

Abbildung 28: Benutzer löschen

Ist ein Benutzer kein Produktmanager und für keinen Datensatz als Bearbeiter eingetragen, könnte dieser Benutzer aus dem System gelöscht werden.

- *Neuen Benutzer anlegen:*

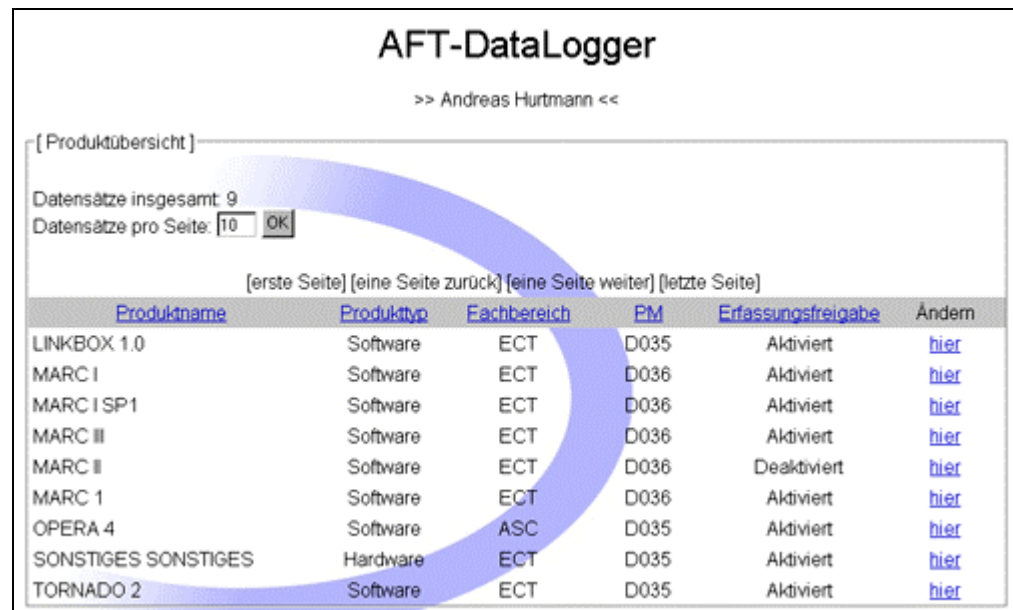


The screenshot displays the AFT-DataLogger web application. At the top, the title 'AFT-DataLogger' is centered, followed by the user's name '>> Andreas Hurtmann <<'. Below this is a section titled '[Neuen Benutzer anlegen]' which contains a form with the following fields: 'Kurzzeichen:' with a small text input, 'Vorname:' with a text input, 'Nachname:' with a text input, 'Passwort:' with a text input, 'E-Mail:' with a larger text input, and 'Administrator:' with a checkbox. At the bottom of the form area are two buttons: 'Benutzer anlegen' and 'Formulardaten zurücksetzen'.

Abbildung 29: Neuen Benutzer anlegen

Hier können neue Benutzer im System angelegt werden. Vergeben werden Kurzzeichen, Vorname, Nachname, Passwort, E-Mail und Administrationsrechte. Der Benutzer kann nur angelegt werden, wenn das vergebene Kurzzeichen noch nicht in der Datenbank existiert.

- *Produktübersicht:*



Produktname	Produkttyp	Fachbereich	PM	Erfassungsfreigabe	Ändern
LINKBOX 1.0	Software	ECT	D035	Aktiviert	hier
MARC I	Software	ECT	D036	Aktiviert	hier
MARC I SP1	Software	ECT	D036	Aktiviert	hier
MARC II	Software	ECT	D036	Aktiviert	hier
MARC II	Software	ECT	D036	Deaktiviert	hier
MARC 1	Software	ECT	D036	Aktiviert	hier
OPERA 4	Software	ASC	D035	Aktiviert	hier
SONSTIGES SONSTIGES	Hardware	ECT	D035	Aktiviert	hier
TORNADO 2	Software	ECT	D035	Aktiviert	hier

Abbildung 30: Produktübersicht

Hier erhält der Administrator eine Übersicht über alle im System angelegten Produkte. Angezeigt werden Produktname, Produkttyp, Fachbereich, Produktmanager (PM) und die Information ob das Produkt für die Freigabe aktiviert ist oder nicht. Unter ‚Ändern‘ kann ein Produkt für die Erfassung aktiviert oder deaktiviert werden. Produkte, die für die Erfassung deaktiviert sind, werden im Erfassungsfeld nicht angezeigt.

- *Neues Produkt anlegen:*

Unter diesem Menüpunkt können neue Produkte angelegt werden. Zuerst wird der Produktname vergeben. Nach verlassen des Feldes wird überprüft, ob dieser Produktname bereits in der Datenbank vorhanden ist, falls ja werden die Daten über den zuständigen Produktmanager übernommen. Falls nicht, müssen diese Informationen eingegeben werden. Die folgenden zwei Abbildungen zeigen beide Möglichkeiten:

AFT-DataLogger

>> Andreas Hurtmann <<

[Neues Produkt anlegen]

Produktname:

Software Version bzw. Gerätetyp:

Produktmanager:

- Kurzzeichen: D035
- Name: Andreas Hurtmann
- E-Mail: a.hurtmann@aft-werdohl.de

Produkttyp: Software

Fachbereich: ECT

Erfassungsfreigabe: ☐

Abbildung 31: Anlegen eines neuen Produktes bei bekanntem Produktnamen

AFT-DataLogger

>> Andreas Hurtmann <<

[Neues Produkt anlegen]

Produktname:

Software Version bzw. Gerätetyp:

Produktmanager:

- Kurzzeichen:
- Name:
- E-Mail:

Produkttyp:

Fachbereich:

Erfassungsfreigabe: ☐

Abbildung 32: Anlegen eines neuen Produktes bei unbekanntem Produktnamen

<p>Produktname:</p>	<p>Nach Verlassen des Feldes wird der eingegebene Text in Grossbuchstaben gewandelt, und in der Datenbank überprüft, ob dieser Name bereits vorhanden ist. Falls ja, werden die Daten dieses Produk-</p>
----------------------------	--

	tes übernommen (Produktmanager, Kurzzeichen, Name, E-Mail, Produkttyp, Fachbereich).
Software Version/Gerätetyp:	Eingabe der Software-Version bzw. der Gerätetyp des Produktes.
Produkttyp:	Auswahl, ob es sich bei diesem Produkt um eine Software oder Hardware handelt.
Fachbereich:	Zuständiger Fachbereich für dieses Produkt. Sollte der Fachbereich noch nicht in der Auswahlbox enthalten sein, so kann in dem nebenstehenden Textfeld der Fachbereich eingegeben werden. Um den Fachbereich auch diesem Produkt zuzuordnen muss über den Button „Hinzufügen“ der Fachbereich der Auswahlbox hinzugefügt werden.

- *Produktdaten ändern:*

AFT-DataLogger

>> Andreas Hurtmann <<

[Produktdaten ändern]

Produktname:

Produktmanager:

- Kurzzeichen:
- Name: Test Person
- E-Mail: d036@aft-werdohl.de

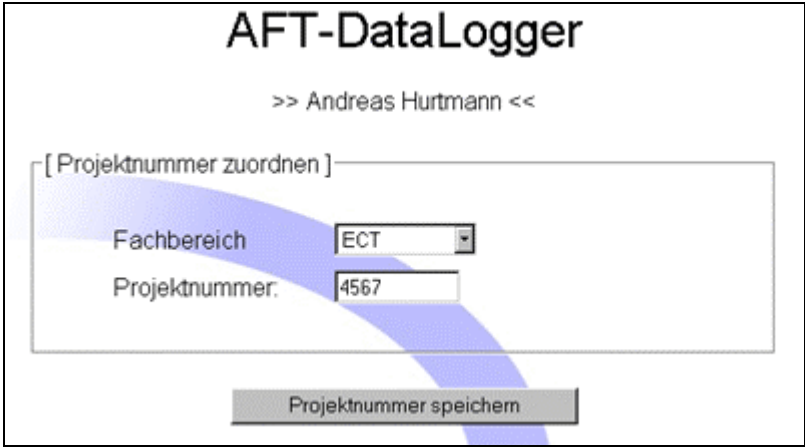
Produkttyp:

Fachbereich

Hier können die Daten über bereits eingepflegte Produkte geändert werden. Nach Auswahl eines Produktes, erhält man Informationen über den zuständigen Produktmanager, den Produkttyp und den Fachbereich. Sollen die Daten geän-

dert werden, klickt man auf den Button „Produktdaten ändern“. Danach kann ein neuer Produktmanager ausgewählt werden und/oder ein neuer Fachbereich zugeordnet werden. Befindet sich der neue Fachbereich nicht in der Auswahlliste, kann über das nebenstehende Eingabefeld ein neuer Fachbereich eingepflegt werden.

- *Projektnummer zuordnen:*



The screenshot displays the 'AFT-DataLogger' application window. At the top, it shows the user name '>> Andreas Hurtmann <<'. Below this is a section titled '[Projektnummer zuordnen]'. Inside this section, there are two input fields: 'Fachbereich' with a dropdown menu showing 'ECT', and 'Projektnummer.' with a text box containing '4567'. A blue arrow points from the 'Projektnummer.' field to a button labeled 'Projektnummer speichern' located below the input fields.

Abbildung 33: Projektnummer zuordnen

Unter diesem Menüpunkt können zu den bereits eingepflegten Fachbereichen Projektnummern gespeichert werden. Diese Projektnummern werden bei der späteren Zuordnung eines Datensatzes zu einer Fehlerkategorie automatisch übernommen.

6.6.3 Produktmanager

Im Hauptmenü des Produktmanagers erhält man einen ersten Überblick über die Anzahl der Gesamtdatensätze, der abgeschlossenen Datensätze, der in Bearbeitung befindlichen Datensätze und der offenen Datensätze.



Abbildung 34: Menu des Produktmanager

- *Übersicht aller Datensätze:*

AFT-DataLogger
 >> Andreas Hurtmann <<

[Übersicht aller Datensätze: Seite 1]

Datensätze insgesamt: 22
 Datensätze pro Seite:

[erste Seite] [eine Seite zurück] [1](#) [2](#) [3](#) [4](#) [eine Seite weiter] [letzte Seite]

Produktname	Software Version / Gerätetyp	E-Datum	KD-Nr	Firma	Status	Anzeigen
OPERA		2001-11-08	067	Pierburg, Neuss	Abgeschlossen	hier
MARC		2001-11-08	065	RWTH, Aachen	Abgeschlossen	hier
TORNADO		2001-11-08	078	CLAAS, Harzewinkel	Abgeschlossen	hier
MARC		2001-11-08	023	Jaguar, U.K.	Abgeschlossen	hier
MARC		2001-11-08	234	AlSIN, Japan	offen	hier
MARC		2001-11-08	345	Trennew, Pforzheim	in Bearbeitung	hier

Abbildung 35: Übersicht über alle Datensätze

In dieser Übersicht erhält man eine Liste aller erfassten Datensätze und ihren Status. Der Status kann „offen“, „in Bearbeitung“ oder „Abgeschlossen“ sein. Über den Link „hier“ unter „Anzeigen“ erhält man sämtliche Daten eines erfassten Datensatzes. Die Anzahl der Daten unterscheidet sich anhand des Status. Bei einem offenen Datensatz erhält man nur eine Übersicht der erfassten Daten, bei einem Datensatz, der sich „in Bearbeitung befindet“ erhält man neben den erfassten Daten auch die Zuordnungsdaten. Bei einem abgeschlossenen Daten-

satz erhält man alle Daten, d.h. erfasste Daten, Zuordnungsdaten und Bearbeitungsdaten.

- *Übersicht der in Bearbeitung befindlichen Datensätze:*

Produktname	Software Version / Gerätetyp	E-Datum	Bearbeiter	KD-Nr	Firma	Details
MARC	I	2001-12-02	D035	228	AISIN, Neu-Isenburg	hier
MARC		2001-11-08	DI	345	Trenow, Pforzheim	hier

Abbildung 36: Übersicht der in Bearbeitung befindlichen Datensätze

In dieser Übersicht sieht man alle Datensätze, die sich aktuell in Bearbeitung befinden. Hier hat man die Möglichkeit, nach einem Klick auf den Link „hier“ unter Details sich die bisher erfassten Daten und die Zuordnungsdaten anzuschauen, sowie die bisherige Zuordnung zu ändern.

- Offene Datensätze zuordnen:

AFT-DataLogger

>> Andreas Hurtmann <<

[Offene Datensätze zuordnen: Seite 1]

Datensätze insgesamt: 10
 Datensätze pro Seite:

[erste Seite] [eine Seite zurück] **1** **2** [eine Seite weiter] [letzte Seite]

Produktname	Software Version / Gerätetyp	E-Datum	Erfasser	KD-Nr	Firma	Zuordnen
MARC	I	2001-12-01	D035	000	AFT, Werdohl	hier
MARC		2001-11-08	D036	345	Trenew, Pforzheim	hier
MARC	I	2001-11-28	D035	234	AISIN, Japan	hier
MARC	III	2001-12-15	D035	228	AISIN, Neu-Isenburg	hier
MARC		2001-11-08	D035	054	AVL List, Graz	hier
MARC		2001-12-02	D035	322	ADTEAM, USA	hier
MARC	III	2001-12-15	DI	234	AISIN, Japan	hier
MARC		2001-11-08	D035	234	AISIN, Japan	hier

Abbildung 37: Übersicht der offenen Datensätze

In dieser Übersicht sieht man alle Datensätze, die bisher noch nicht zugeordnet wurden. Über den Link „hier“ unter „Zuordnen“ erhält man eine Übersicht über die Erfassungsdaten und hat die Möglichkeit, diesen Datensatz zuzuordnen.

[Zuordnen]

Erfaßte Daten:

Allgemeine Daten:	Kundendaten:	Produktdaten:
• Erfasser: D035	• Firma: ADTEAM, USA	• Produktname: MARC
• Meldeart: Telefon	• Kunden-Nr.: 322	• Software Version:
• Datum: 2001-12-02	• Ansprechpartner: Osad	• Revision bzw. Service Pack:
• Zeit: 15:04	• Abteilung:	• Produkttyp: Software
	• Telefon:	• Produktmanager: Andreas Hurtmann
	• Telefax:	• Fachbereich: ECT
	• E-Mail:	• EMail: a.hurtmann@aft-werdohl.de
	• Bemerkung:	• Termin:
		• Beschreibung:

Erfassungsdaten bearbeiten

Zuordnung:

Fehlerkategorie:

Zusatzbeschreibung:

Max. Bearbeitungszeit:

Spätester Endtermin: (Format: yyyy-mm-dd)

Projektnummer:

Zuständiger Bearbeiter:

• Name:

• E-Mail:

Bemerkung:

Abbildung 38: Datensatz zuordnen

Bei der Zuordnung eines Datensatzes sind folgende Angaben erforderlich:

Fehlerkategorie:

Hier wird der Datensatz anhand der Fehlerbeschreibung einer Kategorie zugeordnet. Wurde eine Kategorie ausgewählt, wird die entsprechende Zusatzbeschreibung und die max. Bearbeitungszeit angezeigt. Zusätzlich wird der späteste Endtermin vom Programm automatisch berechnet.

Spätester Endtermin:

Hier wird der späteste Endtermin für die Bearbeitung angezeigt, sobald eine Fehlerkategorie ausgewählt wurde. Dieses Datum kann manuell geändert werden.

Projektnummer:

Die Projektnummer wird automatisch vorgegeben, kann aber manuell geändert werden.

Zuständiger Bearbeiter

Hier wird der Bearbeiter bestimmt, der diesen Datensatz bearbeiten soll. Nach Auswahl eines Kurzzeichens werden die Daten des Bearbeiters angezeigt.

Bemerkung

In diesem Feld können zusätzliche Bemerkungen eingetragen werden, die zur Lösung des Kundenwunsches/AFT-Produktfehlers hilfreich sein könnten.

6.6.4 Bearbeiter

Der Punkt Bearbeiter ist erreichbar, wenn für den eingeloggten User mindestens ein Datensatz zur Bearbeitung ansteht.

Als erstes erhält man eine Übersichtsliste sämtlicher zur Bearbeitung anstehender Datensätze.

AFT-DataLogger							
>> Andreas Hurtmann <<							
[Bearbeiter: Seite 1]							
Datensätze insgesamt: 1							
Datensätze pro Seite: <input type="text" value="10"/> <input type="button" value="OK"/>							
[erste Seite] [eine Seite zurück] [eine Seite weiter] [letzte Seite]							
Produktname	Software Version / Gerätetyp	E-Datum	Erfasser	Enddatum	KD-Nr	Firma	Bearbeiten
MARC	I	2001-12-02	D035	2001-12-30	228	AISIN, Neu-Isenburg	hier

Abbildung 39: Menü des Bearbeiters

In dieser Übersichtsliste werden zunächst die wichtigsten Daten eines jeden Datensatzes angezeigt.

1. Die Produktbezeichnung (Produktname und Software Version bzw. Gerätetyp)
2. Das Erfassungsdatum
3. Das Kurzzeichen des Erfassers
4. Das späteste Enddatum für die Bearbeitung
5. Die Kundennummer des Kunden
6. Die Firmenkurzbezeichnung des Kunden
7. einen Link zum Bearbeiten des Datensatzes

Um einen Datensatz zu bearbeiten klickt man auf den entsprechenden Link in der Zeile des Datensatzes. Die folgende Seite enthält sämtliche Daten des gewählten Datensatzes. Die Seite ist in 3 Bereiche aufgegliedert:

1. Erfassungsdaten
2. Zuordnungsdaten
3. Lösungsdaten

Die folgende Abbildung zeigt die komplette Seite.

AFT-DataLogger
 >> Andreas Hutmair <<

[Lösung eintragen]

Erfasse Daten:

Allgemeine Daten:	Kundendaten:	Produktdaten:
• Erfasser: 0005	• Firma: CLAS, Parawerk	• Produktname: TORNAO
• Meldeart: Telefon	• Kunden-Nr.: 076	• Produkttyp: Software
• Datum: 2001-11-09	• Ansprechpartner: IUI	• Produktmanager: Andreas Hutmair
• Zeit: 08:45	• Abteilung: IT	• Fachbereich: ECT
	• Telefon:	• Email: a.hutmair@ab-werke.de
	• Telefax:	• Thema:
	• E-Mail:	• Beschreibung:
• Bemerkung:		

Zuordnung:

Fehlerkategorie:	Fehler
Zustatsbeschreibung:	Könnte kritische Auswirkungen haben
Max. Bearbeitungszeit:	5 Arbeitstage
Spätester Endtermin:	2001-11-15
Projektnummer:	4507
Zuständiger Bearbeiter:	0005
Name:	Andreas Hutmair
E-Mail:	a.hutmair@ab-werke.de
Bemerkung:	

Fehlerlösung / Erledigung Kundenwunsch:

Fehlerursache:

Fehlerklasse:

Vorbemerkung:

Erfolgskontrolle der Vorbemerkung:

Änderungszustellung:

Datum: 2001-12-01

Abbildung 40: Gesamtansicht der Bearbeitungsseite

Interessant für den Bearbeiter ist hauptsächlich der 3. Teil, denn dort werden die Lösungsdaten eingetragen. Folgende Abbildung hebt diesen Abschnitt hervor.

Fehlerlösung / Erledigung Kundenwunsch:

Fehlerursache:**

Fehlerklasse:**

Sofortmaßnahme:**

Vorbeugemaßnahme:

Erfolgskontrolle der Vorbeugemaßnahme:

Änderungsmitteilung:**

Datum: 2001-12-19

Hinweis: **-markierte Felder sind Pflichtfelder

Daten speichern Formulardaten zurücksetzen Vorgang abschliessen

Abbildung 41: Fehlerlösung/Erledigung Kundenwunsch

Die einzelnen Felder haben dabei folgende Bedeutung:

Fehlerursache:

Hier wird immer die wahre Ursache des Fehlers beschreiben und nicht die Auswirkung.

z.B. Fehlerbeschreibung: Kurzschluss; scheinbare Ursache: Kabel an Metallkante durchgescheuert, Wahre Ursache: Kantenschutz fehlt.

Fehlerklasse:

Hier muss die wahre Fehlerursache einer Fehlerklasse zugeordnet werden. Dies dient zur Klassifizierung der Fehler.

Sofortmaßnahme:

Hier wird die Maßnahme beschrieben, die zur kurzfristigen Fehlerbeseitigung führt.

z.B. Fehlerbeschreibung: Kurzschluss; defektes Kabel erneuert.

Erfolgskontrolle der Sofortmaßnahme:

Beschreibung: wie wurde der Erfolg der Sofortmassnahme bezogen auf die wahre Fehlerursache geprüft.

z.B. Belastungstest des erneuerten und geschützten Kabels an der Metallkante.

Vorbeugemaßnahme:

Hier wird beschrieben, durch welche dauerhafte Maßnahme das Wiederauftreten der wahren Fehlerursache verhindert wird.

z.B. Einführung eines Kantenschutzes in der Fertigung.

Erfolgskontrolle der Vorbeugemaßnahme:

Beschreibung: wie wurde der Erfolg der Vorbeugemaßnahme bezogen auf die wahre Fehlerursache geprüft.

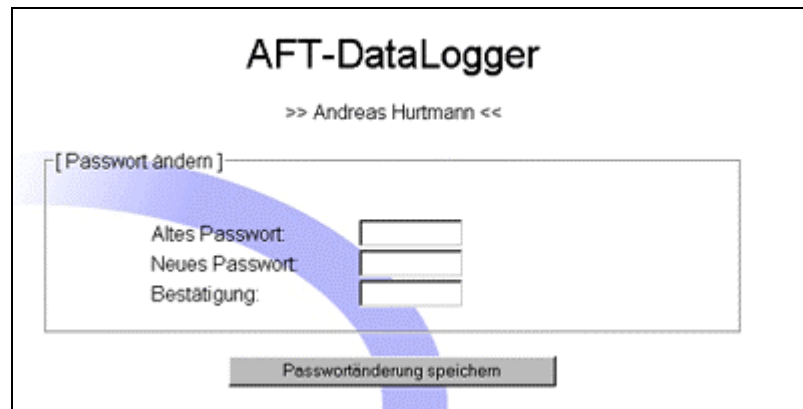
z.B. Belastungstest des erneuerten und geschützten Kabels an der Metallkante.

Änderungsmitteilung:

Was wurde geändert und ab wann (Revision/Datum) ist diese Änderung wirksam.

Sind sämtliche Lösungsdaten eingetragen, kann der Datensatz abgeschlossen werden, dazu klickt man auf den Button „Daten speichern“. Mit dem Speichern (abschließen) des Datensatzes erhält der zuständige Produktmanager eine E-Mail mit dem Hinweis, das dieser Datensatz abgeschlossen wurde.

6.6.5 Passwort ändern



The screenshot shows a web interface for 'AFT-DataLogger'. At the top, it says '>> Andreas Hurtmann <<'. Below this is a menu bar with '[Passwort ändern]' highlighted. The main content area contains three input fields labeled 'Altes Passwort:', 'Neues Passwort:', and 'Bestätigung:'. A blue arrow points from the menu item to the 'Altes Passwort:' field. Below the input fields is a button labeled 'Passwortänderung speichern'.

Abbildung 42: Passwortänderung

Unter diesem Menüpunkt hat man die Möglichkeit sein Passwort zu ändern. Dazu ist zuerst die Eingabe des alten Passwortes nötig. Danach wird das neue Passwort eingegeben, und durch eine wiederholte Eingabe bestätigt. Nach dem Klicken auf den Button „Passwortänderung speichern“ wird die Änderung gespeichert und man erhält einen Hinweis über die erfolgreiche Änderung des Passwortes.

6.7 Datenbankzugriff

Der AFT-DataLogger arbeitet mit dem Datenbanksystem MySQL zusammen. In MySQL wird eine Datenbank „DataLogger“⁴⁹ angelegt, in der der AFT-DataLogger seine Daten speichert. In Abbildung 13 ist die Struktur der Datenbank „DataLogger“ in einem ER-Diagramm abgebildet. Jede Entität entspricht dabei einer Tabelle in der Datenbank „DataLogger“. Folgende Tabellen existieren in dieser Datenbank:

- KDWAFTPF

In dieser Tabelle werden die Erfassungsdaten gespeichert. IM ER-Diagramm ist sichtbar mit welchen anderen Tabellen die Tabelle

„KDWAFTPF“ in Beziehung steht. Diese wären: „Loesung“, „Benutzer“, „Produktversion“ und „Kunde“.

- Loesung

Hier werden die Lösungsdaten eines Datensatzes gespeichert. Diese Tabelle steht in Beziehung mit den Tabellen „Benutzer“ und „KDWAFTPF“.

- Benutzer

In der Tabelle Benutzer stehen die Daten der AFT-Mitarbeiter. Sie steht in Beziehung zu den Tabellen „Loesung“, „Produkt“ und „KDWAFTPF“.

- Produkt

Hier werden die Daten der Produkte gespeichert. Diese Tabelle steht in Beziehung zu den Tabellen „Benutzer“ und „Produktversion“.

- Produktversion

Hier werden die Produktversionen der Produkte gespeichert. Diese Tabelle steht in Beziehung zu den Tabellen „Produkt“ und „KDWAFTPF“.

- Kunde

In der Tabelle „Kunde“ werden die Kundenspezifischen Daten gespeichert. Sie steht in Beziehung zu der Tabelle „KDWAFTPF“.

- Kundendaten

Hier ist immer die aktuelle Kundenliste der Firma AFT gespeichert. Diese Tabelle wird täglich aktualisiert.

- Fehlerklasse

Hier werden die nötigen Daten für die Zuordnung eines Datensatzes zu einer Fehlerklasse gespeichert.

- Fehlerkategorie

Hier werden die nötigen Daten für die Zuordnung eines Datensatzes zu einer Fehlerkategorie gespeichert.

⁴⁹ Im Anhang B stehen die SQL-Statements zur Erzeugung der Tabellenstruktur in der Datenbank „DataLogger“

Die nachfolgende Tabelle enthält eine Auflistung aller Dateien des AFT-DataLogger. Zu jeder Datei wird angegeben, auf welche Tabellen die Datei auf welche Art und Weise zugreift.

Die möglichen vier Zugriffsarten sind folgende:

- SELECT definiert einen Lesezugriff auf eine Tabelle
- INSERT fügt einen neuen Datensatz in eine Tabelle ein
- DELETE löscht einen Datensatz aus einer Tabelle
- UPDATE aktualisiert einen Datensatz in einer Tabelle

Hinweis:

Ein **X** in den letzten vier Spalten bedeutet, dass die entsprechende Zugriffsart angewendet wird, ein – bedeutet keine Anwendung dieser Zugriffsart.

Datei	Tabelle	Select	Insert	Delete	Update
passwort.php	Benutzer	X	-	-	X
login.php		-	-	-	-
login_check.php	Benutzer	X	-	-	-
logout.php		-	-	-	-
menu.php	Benutzer	X	-	-	-
Admin/admin_benutzer.php	Benutzer	X	-	-	-
Admin/admin_menu.php		-	-	-	-
Admin/admin_produkt.php	Produkt	X	-	-	-
	Produktversion	X	-	-	-
	Benutzer	X	-	-	-
Admin/benutzer_aendern.php	Benutzer	X	-	-	-
Admin/benutzer_aendern_check.php	Benutzer	-	-	-	X
Admin/benutzer_anlegen.php		-	-	-	-
Admin/benutzer_anlegen_check.php	Benutzer	X	X	-	-
Admin/benutzer_loeschen.php	Benutzer	X	-	-	-
Admin/benutzer_loeschen_check.php	Benutzer	-	-	X	-
Admin/produktdaten_aendern.php	Produkt	X	-	-	-
	Benutzer	X	-	-	-
Admin/produktdaten_speichern.php	Benutzer	X	-	-	X
	Produkt	X	-	-	-
Admin/produkt_aendern.php	Produkt	X	-	-	-
	Produktversion	X	-	-	-
	Benutzer	X	-	-	-
Admin/produkt_anlegen.php	Produkt	X	-	-	-
	Benutzer	X	-	-	-

Admin/produkt_anlegen_check.php	Produkt	X	X	-	-
	Produktversion	X	X	-	-
	Benutzer	X	-	-	-
Admin/produkt_update_check.php	Produktversion	-	-	-	X
Admin/projekt_nr.php	Produkt	X	-	-	-
Admin/projekt_nr_speichern.php	Produkt	-	-	-	X
Bearbeiter/be_bearbeiten.php	Fehlerkategorie	X	-	-	-
	KDWAFTPF	X	-	-	-
	Loesung	X	-	-	-
	Benutzer	X	-	-	-
	Fehlerklasse	X	-	-	-
Bearbeiter/be_bearbeiten_check.php	Loesung	-	-	-	X
	KDWAFTPF	X	-	-	X
	Benutzer	X	-	-	-
	Produkt	X	-	-	-
Bearbeiter/be_bearbeiten_speichern.php	Loesung	-	-	-	X
Bearbeiter/be_daten.inc	KDWAFTPF	X	-	-	-
	Benutzer	X	-	-	-
	Kunde	X	-	-	-
	Kundendaten	X	-	-	-
	Produktversion	X	-	-	-
	Produkt	X	-	-	-
Bearbeiter/be_menu.php	Loesung	X	-	-	-
	Produkt	X	-	-	-
	KDWAFTPF	X	-	-	-
	Benutzer	X	-	-	-
	Kunde	X	-	-	-
	Kundendaten	X	-	-	-
	Produktversion	X	-	-	-

Bearbeiter/hilfe.html		-	-	-	-
Erfasser/erfasser.php	Kundendaten	X	-	-	-
	Produkt	X	-	-	-
	Produktversion	X	-	-	-
	Benutzer	X	-	-	-
Erfasser/erfasser_check.php		-	-	-	-
Erfasser/erfasser_speichern.php	Benutzer	X	-	-	-
	Produkt	X	-	-	-
	Produktversion	X	-	-	-
	Kunde	-	X	-	-
	KDWAFTPF	-	X	-	-
Produktmanager/daten.inc	KDWAFTPF	X	-	-	-
	Benutzer	X	-	-	-
	Kunde	X	-	-	-
	Kundendaten	X	-	-	-
	Produktversion	X	-	-	-
	Produkt	X	-	-	-
Produktmanager/enddatum.inc		-	-	-	-
Produktmanager/erfassungsaenderung_speichern	Produkt	X	-	-	-
	Produktversion	X	-	-	-
	Kunde	X	-	-	X
	KDWAFTPF	X	-	-	X
Produktmanager/erfassungsdaten_aendern.php	Kundendaten	X	-	-	-
	Produkt	X	-	-	-
	Produktversion	X	-	-	-
	Benutzer	X	-	-	-

Produktmanager/pm_datenubersicht.php	KDWAFTPF	X	-	-	-
	Produkt	X	-	-	-
	Benutzer	X	-	-	-
	Kunde	X	-	-	-
	Kundendaten	X	-	-	-
	Produktversion	X	-	-	-
Produktmanager/pm_daten_anzeigen.php	Fehlerkategorie	X	-	-	-
	Produkt	X	-	-	-
	Benutzer	X	-	-	-
Produktmanager/pm_daten_anzeigen_speichern.php	Benutzer	X	-	-	-
	Loesung	-	X	-	-
	KDWAFTPF	-	-	-	X
Produktmanager/pm_daten_bearbeitung.php	Benutzer	X	-	-	-
	Loesung	X	-	-	-
	Produkt	X	-	-	-
	KDWAFTPF	X	-	-	-
	Kunde	X	-	-	-
	Kundendaten	X	-	-	-
	Produktversion	X	-	-	-
Produktmanager/pm_daten_bearbeitung_anzeigen.php	Fehlerkategorie	X	-	-	-
	KDWAFTPF	X	-	-	-
	Loesung	X	-	-	-
	Benutzer	X	-	-	-
	Fehlerkategorie	X	-	-	-
Produktmanager/pm_daten_bearbeitung_anzeigen_speichern.php	Benutzer	X	-	-	-
	KDWAFTPF	X	-	-	-
	Loesung	-	-	-	X

Produktmanager/pm_daten_uebersicht_anzeigen.php	Fehlerkategorie	X	-	-	-
	KDWAFTPF	X	-	-	-
	Loesung	X	-	-	-
	Benutzer	X	-	-	-
	Fehlerklasse	X	-	-	-
Produktmanager/pm_daten_uebersicht_bearbeiten.php	Fehlerkategorie	X	-	-	-
	KDWAFTPF	X	-	-	-
	Loesung	X	-	-	-
	Benutzer	X	-	-	-
	Fehlerklasse	X	-	-	-
Produktmanager/pm_daten_uebersicht_bearbeiten_speichern.php	Loesung	X	-	-	X
	KDWAFTPF	X	-	-	-
Produktmanager/pm_daten_zuordnen.php	Benutzer	X	-	-	-
	KDWAFTPF	X	-	-	-
	Produkt	X	-	-	-
	Kunde	X	-	-	-
	Kundendaten	X	-	-	-
	Produktversion	X	-	-	-
Produktmanager/pm_menu.php	KDWAFTPF	X	-	-	-

Tabelle 18: Datenbankzugriffe nach Dateien

7 Eingesetzte Software

7.1 PHPMYAdmin

Mittels PHPMYAdmin können ein MySQL-Server komplett oder nur bestimmte Datenbanken in MySQL über einen Webbrowser verwaltet werden. Notwendig für einen reibungslosen Betrieb des Programms ist ein laufender Webserver mit PHP Unterstützung und eine MySQL-Datenbank. Zur Installation muss das heruntergeladene Archiv ins Document Root-Verzeichnis des Web-Servers entpackt zu werden. Im Hauptverzeichnis von PHPMYAdmin befindet sich die Konfigurationsdatei `config.inc.php`. In ihr muss der User inklusive Passwort und der Standort des MySQL-Servers eingetragen werden. Auf der beiliegenden CD befindet sich die Version 2.2.1 rc1. Die aktuelle Version von PHPMYAdmin kann im Internet unter folgender Adresse heruntergeladen werden:

http://phpmyadmin.sourceforge.net/

Diese Version von PHPMYAdmin bietet die folgende Funktionalität:

- Erstellen und Löschen von Datenbanken
- Erstellen, kopieren und modifizieren von Tabellen
- Löschen, editieren und hinzufügen von Feldern
- Ausführen von SQL-Statements und Batch-Dateien
- Schlüsselverwaltung
- Einlesen von Textdateien
- Erzeugen und Einlesen von Tabellen-Dumps
- Export und Import von CSV-Dateien
- Administration von mehreren MySQL-Servern sowie einzelner Datenbanken
- Sprachunterstützung für mehr als 20 Sprachen

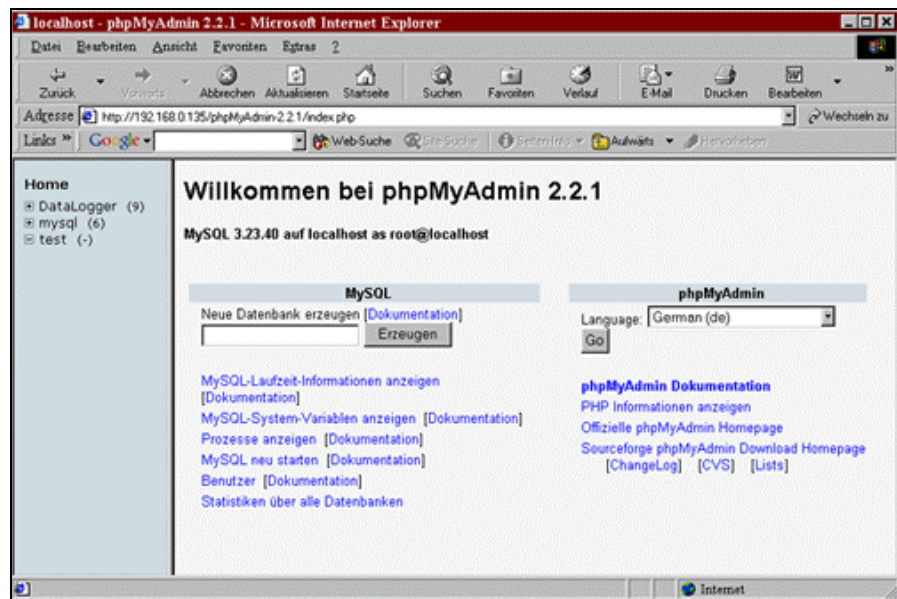


Abbildung 43: Hauptmenü von PHPMyAdmin

Datenbank DataLogger

Tabelle	Aktion	Einträge	Typ	Große
<input type="checkbox"/> Benutzer	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	7	MyISAM	10,4 KB
<input type="checkbox"/> Fehlerkategorie	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	6	MyISAM	2,4 KB
<input type="checkbox"/> Fehlerklasse	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	22	MyISAM	2,8 KB
<input type="checkbox"/> KDWAFTPF	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	23	MyISAM	3,3 KB
<input type="checkbox"/> Kunde	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	24	MyISAM	2,7 KB
<input type="checkbox"/> Kundendaten	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	361	MyISAM	21,1 KB
<input type="checkbox"/> Loesung	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	13	MyISAM	2,5 KB
<input type="checkbox"/> Produkt	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	5	MyISAM	2,2 KB
<input type="checkbox"/> Produktversion	Anzeigen Teilw. anzeigen Einfügen Eigenschaften Löschen Leeren	9	MyISAM	2,2 KB
9 Tabellen	Summe	470	--	49,6 KB

markierte: Oder

Abbildung 44: Tabellenübersicht der Datenbank „DataLogger“

- Textdatei in Tabelle einfügen
- Dump (Schema) der Tabelle anzeigen

☒ Nur Struktur
☐ Struktur und Daten
☐ Nur Daten
☐ CSV-Daten für MS Excel
☐ CSV-Daten :
 Felder getrennt mit :
 Felder eingeschlossen von "
 Felder escaped von \\\n
 Zeilen getrennt mit \\\n

☐ Mit 'DROP TABLE'
☐ Vollständige INSERT's
☐ Erweiterte INSERT's
☐ Tabellen- und Feldnamen in einfachen Anführungszeichen
☐ Senden

Anfangszeile 0 -- Datensätze 7

Abbildung 45: Möglichkeiten des Datenexports einer Tabelle

7.2 PHPed

PHPed ist ein Editor für Windows u.a. für PHP, HTML und Perl. Es besitzt für die erwähnten Programmiersprachen Syntax-Highlighting, was das Programmieren erleichtert, da so eventuelle Flüchtigkeitsfehler frühzeitig erkannt werden. Zudem kann es die erstellten Skripte per FTP⁵⁰ direkt in das Dokumentenverzeichnis des Web-Servers einspielen. PHPed in der verwendeten Version 1.75 Alpha ist ein kostenloses Tool, welches unter Windows läuft. Die aktuelle Version jedoch ist nur noch unter Lizenz zu bekommen. Die kostenlose Version 1.75 Alpha befindet sich auf der beigelegten CD-ROM.

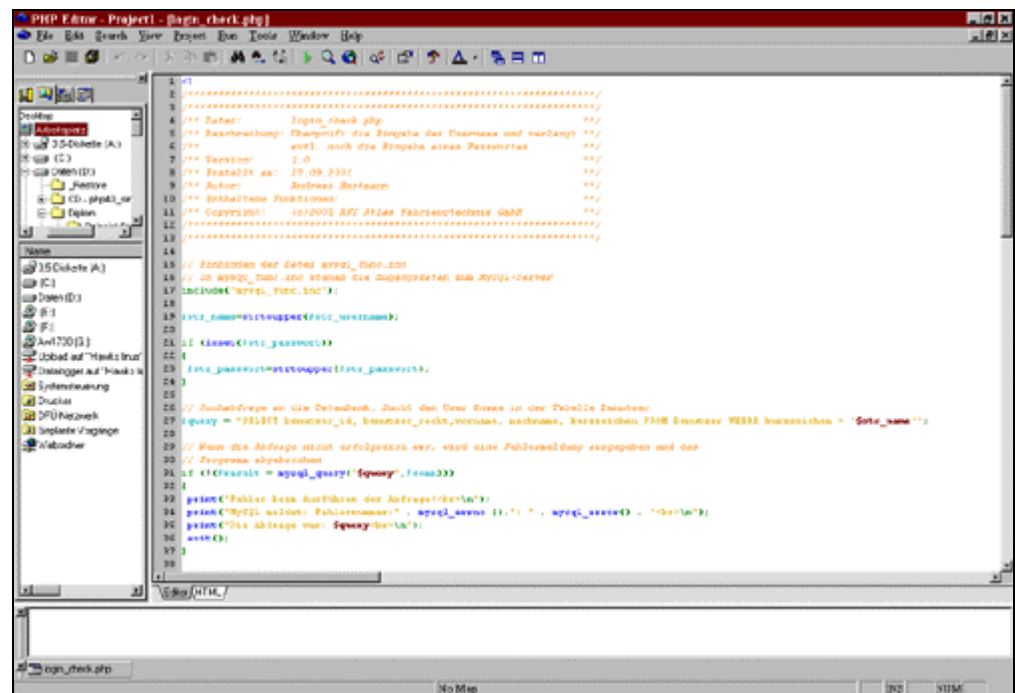


Abbildung 46: PHPEd

7.3 ERwin 3.52

ERwin ist ein Tool, um ER-Diagramme am PC zu erstellen. Es besteht die Möglichkeit, aus den erstellten Entitäten und Beziehungen SQL-Befehle zum Erstellen der Tabellen zu generieren. Außerdem unterstützt es verschiedene Darstellungstypen für die Beziehungen zwischen Entitäten.

⁵⁰ FTP = File Transfer Protocol



Abbildung 47: Startbildschirm von ERwin

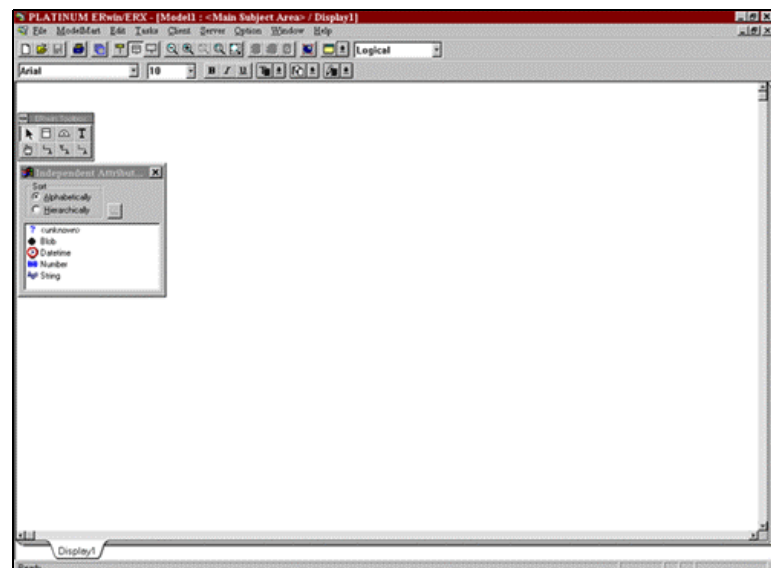


Abbildung 48: Programmoberfläche von ERwin

Unter Optionen → Preferences → Methodology kann die Darstellung für die Beziehungen geändert werden. Die folgende Abbildung zeigt die nötige Einstellung für eine Verwendung der Krähenfussnotation.

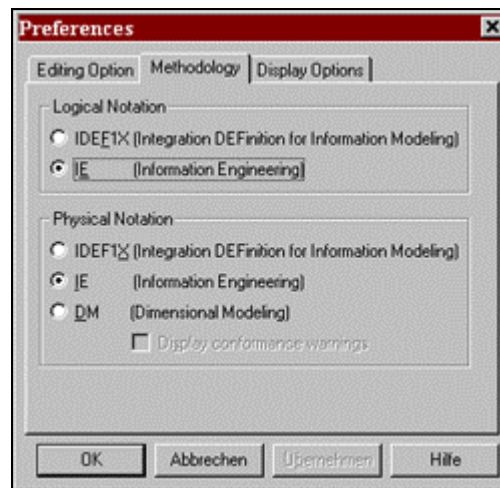


Abbildung 49: Preferences zur Einstellung der Krähenfussnotation

8 Kritische Würdigung und Ausblick

Ziel dieser Diplomarbeit war die Entwicklung und Erstellung einer webbasierten Anwendung zur Erfassung, Verfolgung und Dokumentation von Kundenwünschen und AFT-Produktfehlern. Die entwickelte Anwendung bietet diese Möglichkeit über ein Intranet-System, womit zugleich jedem Mitarbeiter mit Zugriff auf das Intranet die Anwendung zur Verfügung steht. Der AFT-DataLogger befindet sich zum Zeitpunkt der Abgabe der Diplomarbeit noch im Testbetrieb, so dass die endgültige Produkteinführung noch nicht abgeschlossen ist. Nicht Bestandteil dieser Diplomarbeit ist die Auswertung der erfassten Daten. Dieses Feature wird der nächste zu implementierende Schritt sein. Dabei ist es nicht zwingend erforderlich, die in dieser Diplomarbeit verwendeten Techniken zu gebrauchen. Aufgrund der Verwendung einer Datenbank zur Speicherung der Daten kann eine Auswertung auch mit anderen Techniken als PHP, wie z.B. XML, geschehen. Aber aufgrund der ständigen Weiterentwicklung von PHP und seine täglich wachsende Beliebtheit beim Programmieren dynamischer Webseiten, ist eine Umsetzung in dieser Sprache sehr zu empfehlen.

Während der Entwicklung und Testbetrieb erwies sich auch das installierte und konfigurierte LAMP-System als sehr stabil und zuverlässig, so dass der Einsatzserver des AFT-DataLogger ebenfalls ein LAMP-System sein wird.

Zum Schluss möchte ich mich bei den Mitarbeitern der Firma AFT für die gute Unterstützung bedanken, insbesondere gilt der Dank meinen Betreuern Herrn Alfred Schmidt, Herrn Thomas Pusch, Herrn Michael Diem und Herrn Martin Weisner. Bedanken möchte ich mich außerdem noch bei meiner betreuenden Professorin Frau Prof. Dr. Heide Faeskorn-Woyke.

Mein besonderer Dank gilt meinen Eltern und meiner Freundin Tanja, die mich während der ganzen Zeit großartig und verständnisvoll unterstützt haben.

Anhang A – Literaturverzeichnis

- [1] **ATKINSON, LEON (2001):** *Core PHP4 Programmierung*. München. Markt + Technik Verlag
- [2] **BALZERT, HELMUT (1996):** *Lehrbuch der Software-Technik – Software-Entwicklung*. Heidelberg: Spektrum Akademischer Verlag GmbH
- [3] **BALZERT, HELMUT (1998):** *Lehrbuch der Software-Technik – Software-Management, Software-Qualitätssicherung, Unternehmensmodellierung*. Heidelberg: Spektrum Akademischer Verlag GmbH
- [4] **FREEZE, WAYNE (1998):** *Die SQL-Referenz – Das umfassende Nachschlagewerk für Datenbank-Entwickler*. 1. Auflage. Bonn: ITP
- [5] **HEINDL, EDUARD; MAIER, KARIN (1999):** *Der Webmaster – Praktische Realisierung der Internetpräsenz*. Bonn. Addison Wesley Longman GmbH
- [6] **HÖLLER, JOHANN; PILS, MANFRED; ZLABINGER, ROBERT (HRSG.) (1999):** *Internet und Intranet – Auf dem Weg zum Electronic Business*. 2. Auflage. Berlin Heidelberg. Springer-Verlag
- [7] **JAMSA, KRIS; LALANI, SULEIMAN; WEAKLEY, STEVE (1996):** *WEB Programmierung*. Feldkirchen. Franzis-Verlag
- [8] **KOBERT, THOMAS (1999):** *HTML4 – Das Einsteigerseminar*. Kaarst. bhv Verlag
- [9] **KRAUSE, JÖRG (2000):** *PHP Grundlagen und Lösungen*. München Wien. Carl Hanser Verlag
- [10] **KRAUSE, JÖRG (2000):** *PHP4 Grundlagen und Profiwissen*. München Wien. Carl Hanser Verlag
- [11] **MASLAKOWSKI, MARK (2001):** *MySQL in 21 Tagen*. München: Markt + Technik Verlag
- [12] **NEUSS, CHRISTIAN; JOHAN, VROMANS (1996):** *Webmasters Handbuch*. 1. Auflage. Bonn. ITP
- [13] **PAGEL, BERND-UWE; SIX, HANS-WERNER (1994):** *Software Engineering – Band 1 Die Phasen der Softwareentwicklung*. 1. Auflage. Bonn. Addison-Wesley Verlag
- [14] **SEEBOERGER-WEICHSELBAUM, MICHAEL (1999):** *JavaScript - Das Einsteigerseminar*. Kaarst. bhv Verlag
- [15] **SEEBOERGER-WEICHSELBAUM, MICHAEL (2000):** *JavaScript*. 2. Auflage. Kaarst. bhv Verlag
- [16] **STEFFEN, JORGE (1999):** *HTML4 Power*. 5. Auflage. Düsseldorf. SYBEX-Verlag

- [17] **STOLL, ROLF D.; LEIERER, GUDRUN ANNA (2000):** *PHP4 + MySQL*. 1. Auflage. Düsseldorf: DATA BECKER GmbH & Co. KG
- [18] **YARGER, RANDY JAY; REESE, GEORGE; KING, TIM (2000):** *MySQL & mSQL Deutsche Ausgabe*. 1. Auflage. Köln: O'Reilly Verlag GmbH & Co. KG

Anhang B – Quellcode-Auszug

In diesem Anhang möchte ich einige Sourcecode-Auszüge des Programms zeigen. Aufgrund der Menge an Codezeilen kann hier nur ein sehr kleiner Ausschnitt gezeigt werden. Der komplette Sourcecode befindet sich auf der beiliegenden CD-ROM

SQL-Statements zum Anlegen der Datenbankstruktur in MySQL:

```
#  
# Tabellenstruktur für Tabelle `Benutzer`  
#  
  
DROP TABLE IF EXISTS Benutzer;  
CREATE TABLE Benutzer (  
  benutzer_id int(11) NOT NULL auto_increment,  
  kurzzeichen varchar(4) NOT NULL default "",  
  vorname varchar(30) default NULL,  
  nachname varchar(30) default NULL,  
  passwort varchar(10) default NULL,  
  benutzer_recht int(11) default NULL,  
  email varchar(50) default NULL,  
  PRIMARY KEY (benutzer_id),  
  FULLTEXT KEY kurzzeichen (kurzzeichen),  
  FULLTEXT KEY vorname (vorname),  
  FULLTEXT KEY nachname (nachname),
```

```
FULLTEXT KEY email (email)
```

```
) TYPE=MyISAM;
```

```
# -----
```

```
#
```

```
# Tabellenstruktur für Tabelle `Fehlerkategorie`
```

```
#
```

```
DROP TABLE IF EXISTS Fehlerkategorie;
```

```
CREATE TABLE Fehlerkategorie (
```

```
  nr int(11) NOT NULL default '0',
```

```
  beschreibung varchar(50) NOT NULL default "",
```

```
  Zusatzbeschreibung varchar(50) NOT NULL default "",
```

```
  Zieltermin varchar(50) NOT NULL default "",
```

```
  Arbeitstage int(11) NOT NULL default '0',
```

```
  PRIMARY KEY (nr)
```

```
) TYPE=MyISAM;
```

```
# -----
```

```
#
```

```
# Tabellenstruktur für Tabelle `Fehlerklasse`
```

```
#
```

```
DROP TABLE IF EXISTS Fehlerklasse;
```

```
CREATE TABLE Fehlerklasse (
```

```
  kennziffer int(11) NOT NULL default '0',
```

```
  beschreibung varchar(50) NOT NULL default "",
```

```
  gewichtung int(11) NOT NULL default '0',
```

```
  PRIMARY KEY (kennziffer)
```

```
) TYPE=MyISAM;
# -----

#
# Tabellenstruktur für Tabelle `KDWAFTPF`
#

DROP TABLE IF EXISTS KDWAFTPF;
CREATE TABLE KDWAFTPF (
  kdwaftpf_id int(11) NOT NULL auto_increment,
  kunden_id int(11) default NULL,
  benutzer_id int(11) default NULL,
  produkt_id int(11) default NULL,
  produktversion_id int(11) default NULL,
  loesung_id int(11) default NULL,
  version_typ varchar(20) NOT NULL default "",
  datum date NOT NULL default '0000-00-00',
  zeit time default NULL,
  meldeweise varchar(20) default NULL,
  beschreibung text,
  termin varchar(20) default NULL,
  kdwaftpfstatus int(11) default NULL,
  bemerkung text,
  PRIMARY KEY (kdwaftpf_id)
) TYPE=MyISAM;
# -----

#
# Tabellenstruktur für Tabelle `Kunde`
```


#

```
DROP TABLE IF EXISTS Kunde;
CREATE TABLE Kunde (
  kunden_id int(11) NOT NULL auto_increment,
  kdnr int(11) default NULL,
  firma varchar(50) default NULL,
  abteilung varchar(20) default NULL,
  name varchar(30) default NULL,
  telnr varchar(25) default NULL,
  faxnr varchar(25) default NULL,
  email varchar(50) default NULL,
  PRIMARY KEY (kunden_id)
) TYPE=MyISAM;
```

#

Tabellenstruktur für Tabelle `Kundendaten`

#

```
DROP TABLE IF EXISTS Kundendaten;
CREATE TABLE Kundendaten (
  firma varchar(25) NOT NULL default "",
  kdnr int(11) NOT NULL default '0',
  PRIMARY KEY (firma)
) TYPE=MyISAM;
```

```
# -----
```

```
#
```

```
# Tabellenstruktur für Tabelle `Loesung`
```

```
#
```

```
DROP TABLE IF EXISTS Loesung;  
CREATE TABLE Loesung (  
  loesung_id int(11) NOT NULL auto_increment,  
  benutzer_id int(11) default NULL,  
  fehlerursache text,  
  aenderungsmitteilung text,  
  datum date NOT NULL default '0000-00-00',  
  enddatum date NOT NULL default '0000-00-00',  
  fehlerkategorie int(11) NOT NULL default '0',  
  fehlerklasse int(11) NOT NULL default '0',  
  projektnr int(11) NOT NULL default '0',  
  sofortmassnahme text NOT NULL,  
  k_sofortmassnahme text NOT NULL,  
  vorbeugemassnahme text NOT NULL,  
  k_vorbeugemassnahme text NOT NULL,  
  bemerkung text NOT NULL,  
  PRIMARY KEY (loesung_id)  
) TYPE=MyISAM;
```

```
# -----
```

```
#
```

```
# Tabellenstruktur für Tabelle `Produkt`
```

```
#
```

```
DROP TABLE IF EXISTS Produkt;
```

```
CREATE TABLE Produkt (
```

```
    produkt_id int(11) NOT NULL auto_increment,
```

```
    benutzer_id int(11) default '0',
```

```
    produktname varchar(20) NOT NULL default "",
```

```
    produkttyp varchar(20) NOT NULL default "",
```

```
    fachbereich varchar(20) NOT NULL default "",
```

```
    projektnr int(11) NOT NULL default '0',
```

```
    PRIMARY KEY (produkt_id)
```

```
) TYPE=MyISAM;
```

```
# -----
```

```
#
```

```
# Tabellenstruktur für Tabelle `Produktversion`
```

```
#
```

```
DROP TABLE IF EXISTS Produktversion;
```

```
CREATE TABLE Produktversion (
```

```
    produktversion_id int(11) NOT NULL auto_increment,
```

```
produkt_id int(11) default NULL,  
produktversion varchar(20) default NULL,  
produktstatus int(11) NOT NULL default '0',  
PRIMARY KEY (produktversion_id)  
) TYPE=MyISAM;
```

Sourcecode des AFT-DataLogger: login_check.php

```
<?  
/*****  
/*****  
/** Datei:      login_check.php                               **/  
/** Beschreibung: Überprüft die Eingabe des Username und verlangt **/  
/**           evtl. noch die Eingabe eines Passwortes          **/  
/** Version:    1.0                                           **/  
/** Erstellt am: 27.09.2001                                   **/  
/** Autor:      Andreas Hurtmann                             **/  
/** Enthaltene Funktionen:                                   **/  
/** Copyright:  (c)2001 AFT Atlas Fahrzeugtechnik GmbH       **/  
/*****  
/*****  
  
// Einbinden der Datei mysql_func.inc  
// In mysql_func.inc stehen die Zugangsdaten zum MySQL-Server  
include("mysql_func.inc");  
  
$str_name=strtoupper($str_username);
```

```
if (isset($str_passwort))
{
    $str_passwort=strtoupper($str_passwort);
}

// Suchabfrage an die Datenbank. Sucht den User $name in der Tabelle Benutzer
$query = "SELECT benutzer_id, benutzer_recht,vorname, nachname, kurzzeichen FROM Benutzer WHERE kurzzeichen = '$str_name'";

// Wenn die Abfrage nicht erfolgreich war, wird eine Fehlermeldung ausgegeben und das
// Programm abgebrochen
if (!$result = mysql_query("$query",$conn))
{
    print("Fehler beim Ausführen der Abfrage!<br>\n");
    print("MySQL meldet: Fehlernummer:" . mysql_errno() . ": " . mysql_error() . "<br>\n");
    print("Die Abfrage war: $query<br>\n");
    exit();
}

// Wenn kein passwort von login.php übergeben wurde
if (!isset($str_passwort))
{
    // Wenn der User $name nicht in der Tabelle Benutzer vorhanden ist
    if(!mysql_num_rows($result))
    {
        // Weiterleitung an login.php, mit dem Hinweis das das Login fehlerhaft war
        header("Location: login.php?str_login=false&str_username=$str_name");
    }
    // Der User $name ist in der Tabelle Benutzer vorhanden
    else
```

```
{
    $objekt_admin_check = mysql_fetch_object($result);
    // Überprüfen ob der Benutzer als Produktmanager oder als Bearbeiter in der Datenbank vorhanden ist
    $pm_check = mysql_query("SELECT * FROM Benutzer AS A, Produkt AS B where A.benutzer_id = '$objekt_admin_check->benutzer_id' AND
A.benutzer_id = B.benutzer_id;", $conn);
    $be_check = mysql_query("SELECT * FROM Benutzer AS A, Loesung AS B where A.benutzer_id = '$objekt_admin_check->benutzer_id' AND
A.benutzer_id = B.benutzer_id;", $conn);
    // Benutzer ist entweder Administrator, Produktmanager oder Bearbeiter, somit muss er sich mit Passwort
    // identifizieren. Weiterleitung des Users zu login.php mit der Eingabeaufforderung des Passwortes
    if (($objekt_admin_check->benutzer_recht == "1") || (mysql_num_rows($pm_check)) || (mysql_num_rows($be_check)))
    {
        header("Location: login.php?str_login=pw&str_username=$str_name&str_passwort=true");
    }
    // Der Benutzer ist kein Administrator, Produktmanager oder Bearbeiter, somit hat er nur das Recht, Daten zu erfassen
    // Weiterleitung des Users zum Erfassungsformular
    else
    {
        header("Location: Erfasser/erfasser.php?str_username=$str_name&rechte=false&vorname=$objekt_admin_check-
>vorname&nachname=$objekt_admin_check->nachname&name=$objekt_admin_check->kurzzeichen");
    }
}
}
// Ein Passwort wurde von login.php übergeben
else
{
    // Suchabfrage an die Datenbank. Sucht den User $name in Kombination mit dem Passwort $passwort in der Tabelle Benutzer
    $query = "SELECT * FROM Benutzer WHERE kurzzeichen='$str_name' AND password='$str_passwort'";
    // Wenn die Abfrage nicht erfolgreich war, wird eine Fehlermeldung ausgegeben und das
    // Programm abgebrochen
```

```
if (!$result = mysql_query("$query",$conn))
{
    print("Fehler beim Ausführen der Abfrage!<br>\n");
    print("MySQL meldet: Fehlernummer:" . mysql_errno() . ": " . mysql_error() . "<br>\n");
    print("Die Abfrage war: $query<br>\n");
    exit();
}
// Username und Passwort stimmen nicht überein. Der User wird wieder nach login.php verwiesen
if(!mysql_num_rows($result))
{
    header("Location: login.php?str_login=pwfalse&str_username=$str_name&str_passwort=true");
}
// Username und Passwort stimmen. Der User ist identifiziert und kann angemeldet werden
else
{
    $object = mysql_fetch_object($result);
    // Die Session für den identifizierten User wird gestartet
    // und die erforderlichen Variablen für die weitere Identifizierung
    // der Folgeseiten werden registriert. Anschließend wird der User zum Hauptmenu
    // weitergeleitet
    session_start();
    session_register("name");
    session_register("vorname");
    session_register("nachname");
    session_register("gueltigkeit");
    session_register("benutzerrecht");
    session_register("id");
    // Belegen der registrierten Variablen mit Werten
    $name = strtoupper($str_username);
```

```
$vorname = $object->vorname;
$nachname = $object->nachname;
$benutzerrecht = $object->benutzer_recht;
$id = $object->benutzer_id;
// Speichern des Sessionnamens und der benutzer_id
$sessionname = session_name();
$benutzer_id = session_id();
// Mit gültigkeit = true wird eine korrekte Anmeldung identifiziert
$gueltigkeit = true;
// Weiterleitung des Users nach menu.php mit Übergabe des Sessionnamens und der
// eindeutigen benutzer_id
header("Location: menu.php?$sessionname=$benutzer_id");
}
}
?>
```

Sourcecode des AFT-DataLogger: erfasser_speichern.php

```
<?
/*****
/*****
/** Datei:      Erfasser/erfasser_speichern.php          **/
/** Beschreibung: Hier werden die erfaßten Daten nochmals im **/
/**              komprimierter Form aufgelistet. Nach dem drücken **/
/**              auf den OK-Button werden die Daten gespeichert **/
/** Version:    1.0                                     **/
/** Erstellt am: 27.09.2001                             **/
/** Autor:      Andreas Hurtmann                         **/
```



```
/** Enthaltene Funktionen:                                **/
/**      titel.inc      -> PHP                            **/
/**      mysql_func.inc -> PHP                            **/
/**      datum.inc      -> PHP                            **/
/** Copyright:  (c)2001 AFT Atlas Fahrzeugtechnik GmbH    **/
/*****/
/*****/

/// Die mit Sessionname übergebene Session wird wieder eingelesen und initialisiert
session_start();
$sessionname = session_name();
$benutzer_id = session_id();

// Einbinden der Datei mysql_func.inc
// In mysql_func.inc stehen die Zugangsdaten zum MySQL-Server
include("../mysql_func.inc");
include("../datum.inc");

// Störende Leerzeichen werden entfernt
trim($str_ansprechpartner);
trim($str_abteilung);
trim($str_telnr);
trim($str_telfax);
trim($str_email);
trim($str_fehlerbeschreibung);
trim($str_termin);
trim($str_bemerkung);
trim($str_firma);
trim($str_snr);
```

```
if (isset($str_firmaeingabe))
{
    trim($str_firmaeingabe);
}
//Erste Zeichen eines jeden Wortes werden großgeschrieben
$ansprechpartner = ucwords($str_ansprechpartner);
$abteilung = ucwords($str_abteilung);
//Alle Zeichen werden klein geschrieben
$email = strtolower($str_email);
//Setzen von Backslashes vor jedem Sonderzeichen
//quotemeta($str_fehlerbeschreibung);
$fehlerbeschreibung = addslashes($str_fehlerbeschreibung);
// Das Datum wird MySQL gerecht umgewandelt
$erfassungsdatum = $datum;
// Die benutzer_id aus der Tabelle Benutzer wird ausgelesen
if ($rechte == false)
{
    $str_erfasser = $name;
}
$benutzerid = mysql_query("SELECT benutzer_id FROM Benutzer WHERE kurzzeichen = '$str_erfasser'", $conn);
$objekt = mysql_fetch_object($benutzerid);
$int_benutzer_id = $objekt->benutzer_id;
// Die produkt_id wird aus der Tabelle Produkt ausgelesen
$produktid = mysql_query("SELECT produkt_id FROM Produkt where produktname = '$str_produktname';", $conn);
$objekt = mysql_fetch_object($produktid);
$int_produktid = $objekt->produkt_id;
// Die produktversion_id wird aus der Tabelle Produktversion ausgelesen
$produktversionid = mysql_query("SELECT produktversion_id FROM Produktversion AS A, Produkt AS B where A.produktversion = '$str_produktname' AND B.produktname = '$str_produktname' AND A.produkt_id = B.produkt_id", $conn);
```

```
$objekt = mysql_fetch_object($produktversionid);
$int_produkversion_id = $objekt->produktversion_id;

// Konvertierung der Kundennummer in ein INT
settype($int_kdnr, "integer");
if (strlen($str_firmaeingabe) == 0)
{
    $query="INSERT INTO Kunde VALUES('',$int_kdnr,',','$abteilung','$ansprechpartner','$str_telnr','$str_telfax','$email');";
}
else
{
    $query="INSERT INTO Kunde VALUES('','$str_firmaeingabe','$abteilung','$ansprechpartner','$str_telnr','$str_telfax','$email');";
}
if (!($result = mysql_query("$query",$conn)))
{
    print("Fehler beim Ausführen der Abfrage!<br>\n");
    print("MySQL meldet: Fehlernummer:" . mysql_errno() . ": " . mysql_error() . "<br>\n");
    print("Die Abfrage war: $query<br>\n");
    exit();
}
$insert_id = mysql_insert_id($conn);
$query_kdwaftpf="INSERT INTO KDWAFTPF VAL-
UES(',$insert_id','$int_benutzer_id','$int_produktid','$int_produkversion_id','$str_snr','$erfassungsdatum','$zeit','$str_meldeform','$fehlerbeschre
ibung','$str_termin','0','$str_bemerkung');";
if (!($result_kdwaftpf = mysql_query("$query_kdwaftpf",$conn)))
{
    print("Fehler beim Ausführen der Abfrage!<br>\n");
    print("MySQL meldet: Fehlernummer:" . mysql_errno() . ": " . mysql_error() . "<br>\n");
    print("Die Abfrage war: $query_kdwaftpf<br>\n");
}
```

```
exit();
}
// Die E-Mail wird verschickt
$message="$str_erfasser hat einen neuen Kundenwunsch/AFT-Produktfehler erfaßt. \n\nBitte schauen Sie im System nach und Klassifizieren Sie diesen Datensatz, und weisen Sie ihm einen Bearbeiter zu";
$header="FROM: \"AFTDataLogger\" <root@werdohl.aft-werdohl.de>\n";
mail("$str_pmemail", "Neuer Datensatz im AFT-DataLogger", $message, $header);
$str_msg = "Die Daten wurden erfolgreich in der Datenbank gespeichert, und der zuständige Produktmanager wurde per E-Mail informiert";
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
<head>
<title>Erfasser Check</title>
<link href="../aft.css" rel="stylesheet" type="text/css">
<base target="Haupt">
</head>
<body background="../img/Ovalfadebg7h.jpg">
<!-- Beginn Menuzeile -->
<table border="0" width="100%" cellpadding="4" cellspacing="0">
<tr>
<td align="left">
<?
if ($rechte != "false")
{
print("<a href='../menu.php?sessionname=$benutzer_id'>Hauptmenu</a> - Kundenwunsch/AFT-Produktfehler erfassen</td>\n");
}
else
{
```

```
print("<a  
href=\"erfasser.php?username=$name&rechte=false&vorname=$vorname&nachname=$nachname&name=$name\">Kundenwunsch/AFT-  
Produktfehler erfassen</a></td>");  
}  
?>  
<td align="right"><a href=" ../logout.php?$sessionname=$benutzer_id">LOGOUT</a></td>  
</tr>  
</table>  
<hr>  
<!-- Ende Menuzeile -->  
<!-- Beginn Überschrift -->  
<?  
require("../titel.inc");  
?>  
<!-- Ende Überschrift -->  
<br>  
<!-- Beginn der äußeren Tabelle -->  
<table border="0" width="60%" align="center" cellspacing="0" cellpadding="4">  
<tr>  
<td>  
<fieldset>  
<legend>[ Kundenwunsch/AFT-Produktfehler erfassen ]</legend>  
<div align="center">  
<!-- Beginn der inneren Tabelle -->  
<table border="0" cellpadding="4" cellspacing="0">  
<colgroup>  
<col width="10%">  
<col width="10%">  
<col width="20%">
```

```
<col width="30%">
<col width="10%">
</colgroup>
<tr>
  <td align="left" colspan="5">&nbsp;</td>
</tr>
<tr>
  <td align="center" colspan="5"><? print("$str_msg"); ?></td>
</tr>
<tr>
  <td align="left" colspan="5">&nbsp;</td>
</tr>
</table>
</div>
</fieldset>
</td>
</tr>
</table>
</body>
</html>
```

Anhang C – Inhalt der CD-ROM

/Diplomarbeit

In diesem Verzeichnis befindet sich die vorliegende Diplomarbeit im Word2000 – sowie im PDF-Format und das Pflichtenheft des AFT-DataLogger.

/AFT-DataLogger

Hier befindet sich der Sourcecode des AFT-DataLogger. In entpackter und in gepackter Version. Außerdem ist hier der Dump der Datenbank DataLogger zu finden.

/Sourcen

In diesem Verzeichnis befinden sich die Programme der eingesetzten Software: PHP4.06, MySQL 3.23.40 und Apache 1.3.20.

/Programme

In diesem Verzeichnis befinden sich die lizenzfreien Programme PHPMyAdmin und PHPed.

Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Gummersbach, den 16.01.2002

Andreas Hurtmann